НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ І
СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ

«На правах рукопису»
УДК 004.3

«До захисту допущено»
Завідувач кафедри СПСКС

_____ Віталій РОМАНКЕВИЧ
(підпис) (ім'я, прізвище)
"___"_____2020р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 123 Комп'ютерна інженерія (Спеціалізовані комп'ютерні системи)

на тему: Система онлайн-голосування на базі технології Blockchain

Виконав: студент ІІ курсу, групи КВ-93мп

Довжик Дмитро Вікторович _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник доцент, к.т.н., Потапова К. Р. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант з нормоконтролю доцент, с.н.с.,к.т.н. Юлія БОЯРІНОВА _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2020 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – другий (магістерський)

за освітньо-професійною програмою

Спеціальність 123 Комп'ютерна інженерія (Спеціалізовані комп'ютерні системи)

ЗАТВЕРДЖУЮ
Завідувач кафедри СПСКС
_____ Віталій РОМАНКЕВИЧ
(підпис) (ініціали, прізвище)
«___»_____2020_р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Довжик Дмитро Вікторович
(прізвище, ім'я, по батькові)

1. Тема дисертації Система онлайн-голосування на базі технології Blockhain ,

науковий керівник дисертації доцент, к.т.н., Потапова Катерина Романівна ,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «12» листопада 2020 р. №3298-С

2. Термін подання студентом дисертації 10 грудня 2020 р.

3. Об’єкт дослідження Ефективність технології Blockchain в системі онлайн-голосування

4. Предмет дослідження Система онлайн-голосування на базі технології Blockchain

5. Перелік завдань, які потрібно розробити Огляд існуючих рішень, аналіз недоліків існуючих рішень, вибір базової ідеї, створення та обґрунтування протоколу роботи, моделювання спроектованої системи, аналіз роботи.

6. Перелік ілюстративного матеріалу - презентація

7. Перелік публікацій - 2 тез, 2 наукові статті в фахових виданнях

8. Дата видачі завдання 5 листопада 2019 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Затвердження теми	11.10.2019	
2	Збір та дослідження літератури	19.09.2020	
3	Аналіз існуючих рішень	23.09.2020	
4	Зміст та вступ	25.09.2020	
5	Розробка програмної моделі	05.10.2020	
6	Реферат	09.10.2020	
7	Перший розділ	09.10.2020	
8	Другий розділ	23.10.2020	
9	Третій розділ	04.11.2020	
10	Четвертий розділ	16.11.2020	
11	Редагування та перевірка роботи	20.11.2020	

12	Попередній розгляд магістерської дисертації на кафедрі	27.11.2020	

Студент _____ Дмитро ДОВЖИК
(підпис)

Науковий керівник дисертації _____ Катерина ПОТАПОВА
(підпис)

РЕФЕРАТ

Актуальність теми. Використання технології блокчейн набуває свою популярність для задач, де головними критеріями є забезпечення надійності та захисту даних. Голосування на виборах до різних органів державної влади є одним з методів волевиявлення суспільства. Альтернативою традиційному голосуванню є електронне або онлайн-голосування. На сьогодні багато країн розглядають можливість впровадження систем онлайн-голосування з метою удосконалення різних аспектів виборчого процесу.

Основна проблема існуючих систем онлайн-голосування полягає у централізації даних, тобто усі дані зберігаються на одному сервері і підрахунок голосів здійснюється тому ж сервері, тому такі системи є вразливими до зовнішнього втручання. Розподілені системи можуть вирішити цю проблему. Провівши дослідження розподілених систем різного виду, для задачі онлайн-голосування було обрано технологію Blockchain.

Об'єктом дослідженняє процес онлайн-голосування на базі технології Blockchain.

Предметом дослідження є методи захисту даних та алгоритми досягнення консенсусу у мережі Blockchain.

Метою роботи є розробка системи для забезпечення надійності та захищеності процесу голосування та його результатів, а також пришвидшення та спрощення, з використанням можливостей мережі Blockchain.

Для досягнення мети дослідження поставлено і вирішено такі завдання:

- дослідження структури та принципів побудови систем онлайн-голосування;
- розробка програмної моделі мережі Blockchain для моделювання онлайн-голосування;
- опис роботи моделі та аналіз отриманих результатів.

Методи дослідження: методи досягнення несуперечливості даних та методи захисту інформації в мережі Blockchain.

Наукова новизна одержаних результатів роботи полягає у наступному:

- запропоновано метод забезпечення надійності та захищеності систем онлайн-голосування з використанням технології Blockchain, за допомогою шифрування даних та розподіленого збереження даних;
- розроблено програмне забезпечення для моделювання роботи системи онлайн-голосування на базі мережі Blockchain.

Проведене дослідження дає можливість виконувати симуляцію роботи мережі Blockchain для онлайн-голосування.

Практична цінність у можливості використання розробленної системи для різних видів голосувань:

- президентські вибори;

- вибори у парламент;
- місцеві вибори.

Публікації:

- Dovzhyk D., Potapova K., Online-voting system based on blockchain technology // Priority directions of science and technology development. Abstract of the 2-nd International scientific and practical conference. SPC "Sci-conf.com.ua". Kyiv, Ukraine. 2020. Pp. 248-249;
- Довжик Д. В., Потапова К. Р., Система онлайн-голосування на базі технології блокчейн // Прикладна математика та комп'ютинг. Матеріали XIII конференції молодих вчених ПМК-2020. м. Київ, 18-20 листопада, 2020;
- Довжик Д. В., Потапова К. Р., Використання національних засобів криптографічного захисту інформації для шифрування блоків блокчейну // Науковий журнал «Вчені записки Таврійського національного університету імені В. І. Вернадського». Том 31 (70) №6, 2020;
- Довжик Д. В., Потапова К. Р. Система онлайн-голосування на базі технології Blockchain з використанням національних стандартів шифрування // Міжнародний науковий журнал «Інтернаука». — 2020. — №18.

Структура та обсяг роботи. Магістерська дисертація складається зі вступу, чотирьох розділів, висновків та додатків.

У вступі представлена загальна характеристика роботи, описана постановка задачі та запропоноване використання технології Blockchain для системи онлайн-голосування, обґрунтована актуальність роботи.

У першому розділі описані та порівнянні між собою існуючі системи електронного голосування.

У другому розділі описана предметна область задачі, визначені основні функції та вимоги до системи, наведені основні прецеденти користування з системою, описано проектування графічного інтерфейсу системи.

У третьому розділі обґрунтовано вибір технологій та бібліотек для реалізації додатку, наведено обґрунтування вибору мови програмування для розробки системи, описана архітектура модуля клієнта та сервера.

У четвертому розділі наведений аналіз можливості використання розробленого проекту та опис ідеї стартап-проекту, визначення базової стратегії розвитку та позиціонування стартап-проекту.

У висновках стисло наводяться результати розробки та досліджень.

Ключові слова: онлайн-голосування, Blockchain, захист інформації, технологія блокчейн, консенсус, надійність даних, Proof of work.

ABSTRACT

Topicality. The use of blockchain technology is gaining popularity for tasks where the main criteria ensuring the reliability and data protection. Voting in elections to various government bodies is one of the methods of expressing the freedom of society. An alternative to traditional voting is electronic or online voting. Today, many countries are considering introducing online voting systems to improve various aspects of the electoral process.

The main problem with existing online voting systems is the centralization of data, since all data is stored on one server and the counting of votes is carried out on the same server, so such systems are vulnerable to external interference. Distributed systems can solve this problem. After researching distributed systems of various types, Blockchain technology was chosen for the online voting task.

The object of the study is the process of online-voting based on Blockchain technology.

The subject of the study is data protection methods and algorithms for achieving consensus in the Blockchain network.

The purpose and objectives of the study. The purpose of the Master's thesis is to ensure the reliability and security of the voting processes and its results, as well as to accelerate and simplify, using the capabilities of Blockchain network.

To achieve the goal, the following tasks were set and solved:

- study of the structure and principles of building online voting systems;
- development of a software model of the Blockchain network for modeling online voting simulation;
- description of the model and analysis of the results.

Research Methods. To achieve the goals set in the master's thesis, methods of achieving data consistency and methods of protecting information in the Blockchain network were used. The scientific novelty of the work is as follows:

- a method was proposed to ensure the reliability and security of online-voting systems using Blockchain technology, using data encryption and distributed data storage;

- a software product has been developed to simulate the operation of the online-voting system based on the Blockchain network.

The conducted research makes it possible to simulate the work of the Blockchain network for online-voting.

Practical value. The developed system can be used for various types of voting:

- presidential elections;
- parliamentary elections;
- local elections.

Posts:

- Dovzhyk D., Potapova K., Electronic voting system based on blockchain technology // Priority directions of science and technology development. Abstract of the 2-nd International scientific and practical conference. SPC "Sci-conf.com.ua". Kyiv, Ukraine. 2020. Pp. 248-249;
- Dovzhyk D., Potapova K., Online-voting system based on blockchain technology // Applied mathematics and computing. Abstract of XIII conference of young scientists PMK-2020. Kyiv, November 18-20, 2020;
- Dovzhyk D., Potapova K., Use of national facilities of cryptographic protection of information for encryption of blockchain blocks. Scientific journal “Scientific notes of Taurida VI Vernadsky National University. Series: Technical Sciences” Volume 31(70) № 6, 2020;
- Dovzhyk D., Potapova K., Online-voting system based on Blockchain technology using encryption standards // International scientific journal “ Internauka”». — 2020. — №18.

Structure and scope of work. The master's dissertation consists of an introduction, four chapters, conclusions and appendices.

The introduction presents the general characteristics of the work, describes the problem statement and proposes the use of Blockchain technology for the online voting system, substantiates the relevance of the work.

The first section describes and compares existing electronic voting systems.

The second section describes the subject area of the problem, defines the main functions and requirements for the system, provides the main precedents for use with the system, describes the design of the graphical interface of the system.

The third section substantiates the choice of technologies and libraries for the implementation of the application, provides justification for the choice of programming language for system development, describes the architecture of the client module and server.

The fourth section provides an analysis of the possibility of using the developed project and a description of the idea of a startup project, determining the basic development strategy and positioning of the startup project.

The conclusions summarize the results of development and research.

Keywords: Online-voting, Blockchain, information protection, blockchain technology, consensus, data reliability, Proof of work.

ЗМІСТ

1

4

6

8

8

8

10

14

15

18

12 Огляд існуючих рішень20

1.2.1 Сучасні системи електронного голосування20

1.2.2 Огляд існуючих реалізацій21

23

24

24

27

28

28

29

30

31

31

32

2.6.1 Proof of work – алгоритм досягнення консенсусу34

35

35

36

36

36

Error! Bookmark not defined.

38

38

39

39

39

41

43

43

3.1.1 Вибір мови програмування та інструменту для розробки графічного інтерфейсу користувача43

49

51

52

54

55

58

58

60

a. Маркетинговий аналіз60

b. Технологічний аудит ідеї проекту60

c. Аналіз ринкових можливостей запуску стартап-проекту61

4.4. SWOT-аналіз65

4.5. Розроблення маркетингової програми стартап-проекту69

73

74

75

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

Віджет	— графічний елемент або модуль інтерфейсу.
Віртуальна машина	— модель обчислювальної машини, яка створен шляхом віртуалізації обчислювальних ресурсів (оперативної пам'яті, пристроїв збереження та вводу/виводу інформації).
Графічний інтерфейс користувача	— (англ. Graphical user interface, GUI) — тип інтерфейсу, що дозволяє користувачам взаємодіяти з пристроями через графічні зображення та візуальні вказівки.
Клієнт	— персональний комп'ютер або робоча станція, яка запрошує інформацію у сервера.
Прикладний програмний інтерфейс	— (англ. Application Programming Interface, API) — інтерфейс взаємодії, що є набором класів, функцій, який представлений операційною системою чи додатком для використання у інших програмний додатках.
Сервер	— комп'ютер (як правило, потужна робоча станція), що зберігає інформацію, з якою працюють клієнти.
Фреймворк	— заготовки, шаблони для програмної платформи, що визначають архітектуру програмної системи.
Bitcoin	— перша криптовалюта яка базується на технології блокчейн.
DOM	— (англ. Document Object Model, DOM) — об'єктна модель документа прикладного програмного інтерфейсу для роботи

	зі структурованими документами.
HTML	— (англ. HyperText Markup Language) — мова розмітки гіпертекстових документів — стандартна мова розмітки веб-сторінок.
HTTP	— (англ. HyperText Transfer Protocol) — прикладний протокол передачі даних в мережі, використовується для отримання інформації з веб-сайтів.
Microsoft .NET Framework	— платформа, що створена компанією Microsoft, для розробки та виконання клієнтський та серверних програм і розроблена для роботи під операційними системами сімейства Microsoft Windows.
P2P	— (англ. Peer-to-peer) — архітектура мережі системи в якій усі вузли рівноправні.
Sdk	— (англ. Software development kit) — комплект бібліотек для розробки програмного забезпечення.

ВСТУП

Технологія Blockchain набуває свою популярність для завдань, де головними критеріями є забезпечення надійності та захисту даних. Назва складається з слів «Block» та «Chain», що дослівним перекладом є «ланцюг блоків». Технологія Blockchain була винайдена в 2008 році і розроблена людиною або групою людей під псевдонімом Сатоші Накамото. Спочатку блокчейн був базою для Bitcoin – одна з найбезпечніших цифрових фінансових систем. Однак технологія Blockchain – це лише інструмент для зберігання та передачі даних, і теоретично вона може бути використана в будь-якій галузі електронної комерції.

Blockchain також активно обговорюють як можливість вдосконалення та спрощення певних державних операцій. Смарт-контракти на основі Blockchain – це контракти, які можуть бути частково або повністю виконані без взаємодії людини. Однією з головних цілей смарт-контрактів є автоматизоване депонування. Під час обговорення співробітники Міжнародного Валютного Фонду повідомляли, що смарт-контракти на базі технології Blockchain можуть зменшити моральний ризик та оптимізувати використання контрактів взагалі.

Банки дуже зацікавлені в технології Blockchain через її потенціал для пришвидшення системи врегулювання у місцевих відділеннях.

Для приладу, такий банк, як UBS, відкриває нові дослідницькі лабораторії, присвячені технології Blockchain, щоб дослідити, як можна використовувати Blockchain в фінансових послугах для підвищення ефективності та зменшення витрат.

Існує ряд компаній та галузевих організацій, що працюють над використанням технології Blockchain в логістиці ланцюгів поставок та управлінні ланцюгами поставок.

Альянс Blockchain in Transport Alliance (BiTA) спрямовує свої ресурси на розробку відкритих стандартів для управління ланцюгами поставок.

Технологічна компанія Everledger є одним з перших клієнтів служби трекінгу на базі блокчейну IBM.

Компанії Walmart та IBM співпрацюють для тестування систем, що працюютьна основі технології Blockchain, для моніторингу ланцюга поставок товарів. Усі вузли блокчейну керуються Walmart і знаходяться в хмарному сховищі IBM.

Платформа Hyperledger Grid розробляє відкриті компоненти та утиліти для рішень управління ланцюгів поставок з використанням технології Blockchain.

Але найбільша дискусія ведеться з приводу використання технології Blockchain для демократичних виборів. Голосування завжди було складним процесом, де високий ступінь довіри перебуває в руках невеликої кількості людей. Поки існувала демократія, голосування було централізоване, з часом довіра людей до системи руйнувалася, що призвело до апатичного погляду виборців у багатьох різних культурах. У країнах, що розвиваються, часто дії уряду щодо голосування піддаються сумніву. Новий метод голосування може принести справедливі результати і відповідно значно поліпшити життя мешканців.

Чому технологія Blockchain розглядається як рішення для чесного голосування? Тому що, технологія Blockchain вирішує головну проблему традиційного голосування – централізацію. Основна ідея Blockchain це децентралізація збереження даних. У кожного учасника ланцюга є своя копія даних, всі зміни додаються поступово і поширюються усім учасникам ланцюга. Інформація в ланцюзі відкрита для кожного. Тому практично неможливо підробити результат голосування, оскільки це вимагало б злому усіх користувачів без винятку та зміни усіх копій даних.

З вищевказаних причин метою даної роботи є створення системи онлайн-голосування на базі технології Blockchain. На сьогоднішній день майже кожен громадянин України має доступ до Інтернету, тому онлайн-система голосування може значно оптимізувати витрати виборчого процесу та пришвидшити його, одночасно зробити його прозорішим. Завдяки такій системі можна було б здійснити будь-яке голосування кількома натисканнями на смартфоні чи комп’ютері, починаючи від голосувань на місцевих виборах та закінчуючи голосуванням за президента держави.

РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1 Технологія Blockchain

1.1.1 Особливості технології

Blockchain (chain of block) — це розподілена база даних, в якій пристрої зберігання даних не підключені до одного загального сервера. Ця база даних зберігає постійно зростаючий список упорядкованих незмінних записів даних, названих блоками, які зберігаються на всіх вузлах мережі в одному і тому ж вигляді. Кожен із цих блоків даних (тобто блок) захищений за допомогою криптографічного алгоритму і пов’язаний з іншими (тобто блоки складають ланцюг або дерево). Блок містить мітку часу і посилання на попередній блок геш дерева. Для запису нового блоку, необхідно послідовне зчитування інформації про старі блоки. Така розподілена база даних була закладена в основу цифрової валюти Bitcoin та інших, де використовується як книга обліку для всіх виконаних транзакцій.

Blockchain-мережа не має центральних повноважень – саме це є визначенням демократизованої системи. Оскільки це загальний і незмінний реєстр, інформація в ньому відкрита для кожного. Тому, все, що побудовано на блокчейні, за своєю природою є прозорим, і кожен учасник несе відповідальність за свої дії[1].

Функціонування блокчейну відбувається в режимі P2P (комп'ютерна мережа, де всі учасники рівноправні). Всі операції проводяться між суб'єктами безпосередньо. А здійснюються вони за рахунок того, що всі учасники підключені до однієї мережі — блокчейн.

Кожна транзакція ланцюга блокчейну захищена цифровим підписом, що підтверджує її автентичність і засвідчує її. Через використання шифрування і цифрових підписів, дані, які зберігаються у мережі блокчейну, є захищеними від зовнішнього втручання і не можуть бути відредаговані.

Технологія Blockchain дозволяє всім учасникам мережі досягти узгодженності, що називають досягнення консенсусу. Всі дані, збережені у мережі блокчейн, записуються цифровим способом, тому відображені у загальному реєстрі, який доступний для всіх учасників мережі. Це виключає будь-

який шанс на шахрайську діяльність або дублювання транзакцій без потреби третьої сторони.

Блокчейн не несе витрат на транзакції. Блокчейн - це простий і прозорий спосіб передачі інформації з точки А в точку В повністю безпечний і автоматизований. Одна сторона, що ініціює процес транзакції, створює блок. Цей блок перевіряють усі комп'ютери, що розподілені по мережі. Перевірений блок додається до кінця ланцюга, не тільки створюючи унікальний запис, але й створюючи унікальний запис з унікальною історією. Підробка одного запису у ланцюзі означала б підробку кожного екземпляру цього ланцюга. Це майже неможливо.

Криптовалюта Bitcoin використовує таку модель для грошових транзакцій, але її можна використати і іншими варіаціями у інших галузях [4].

Для прикладу можна уявити залізничну компанію. Пасажири купують квитки у мобільному додатку чи в мережі Інтернеті. Компанія бере передплату на обробук транзакції з кредитних карток. Завдяки блокчейну залізничні оператори можуть не лише заощадити на обробці платежів з кредитної картки, але й може перемістити весь процес оформлення квитків у блокчейн. Клієнт та залізнична компанія – дві сторони угоди. Дані про квиток – це блок, який буде доданий у блокчейн квитків. Подібно до того, грошова транзакція теж унікальна, перевіряється незалежно і є непідробним записом блокчейну (як Bitcoin), таким може бути і пасажирський квиток.

Але найголовніше тут те, що це безкоштовно. Blockchain може не тільки переказувати та зберігати кошти, але й заміняти всі процеси та бізнес-моделі, які покладаються на стягнення комісії за транзакції. Або будь-яка інша операція між двома сторонами [5. С. 121].

Навіть таким компаніям, як AirBnB і Uber, загрожує технологія Blockchain. Необхідно просто закодувати інформацію про транзакцію за оплату ночівлі або за проїзд у таксі, і виникає цілком безпечний спосіб отримати послуги, які не

потребують від компаній участі та порушують їхню бізнес-модель, а не так давно ці компанії кидали виклик традиційній економіці.

Тому це не тільки витіснення посередника, але й загалом виключення необхідності у таких платформах.

Оскільки транзакції блокчейну безкоштовні, можна стягувати невелику плату за перегляд відео чи читання статті, наприклад 0.001% від суми [5. С. 223]. Тоді можна стягувати будь-яку суму і не турбуватись, що третя сторона забере частину прибутку за надання своїх послуг.

Також на електронні книги може бути встановлена система, яка базується на Blockchain. Книги будуть розповсюджуватись у закодованому вигляді, а підтверджена транзакція Blockchain передасть гроші автору книги та розблокує її для читача,замість того, щоб платити відсоток книжковому магазину. Тоді автор отримуватиме всю суму за свій твір, а не тільки малі відсотки від магазину. В такому випадку магазин книг Amazon стає непотрібним [5. С. 223-224].

У фінансовій сфері технологія Blockchain змінить процеси фондових бірж, страхування і придбання позик. Технологія Blockchain усуне банківські рахунки та практично всі послуги, які пропонують банки, коли переваги технології Blockchain з надійним реєстром даних, без комісій за транзакції, будуть широко зрозумілі та поширені. Фінансова система працює за рахунок того, що бере невелику частину грошей за свої послуги здійснення транзакції [5. С. 224].

1.1.2 Структура блоку та етапи роботи мережі

Кожен блок із ланцюга блокчейна містить адресу, дату та час створення транзакції, інформацію, геш і список попередніх транзакцій. Приклад ланцюга блокчейну, що складається з трьох блоків зображений на рис. 1.1.

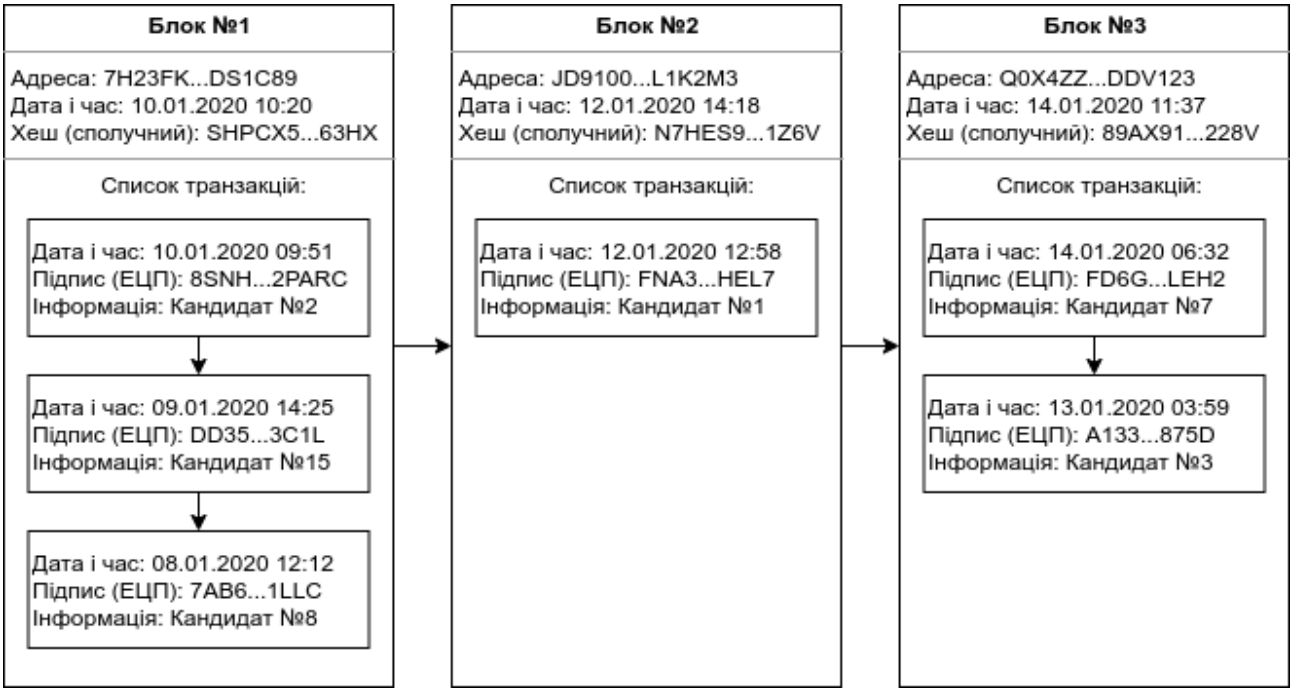

Рисунок 1.1 – Приклад ланцюга з трьох блоків

Структура блоку:

1) адреса – відкритий ключ, генерований асиметричним алгоритмом шифрування, на основі приватного ключа користувача;
2) дата і час – час створення транзакції (блоку);
3) геш – обчислюється за допомогою функції гешування від адреси попереднього блоку і суми гешів всіх транзакцій поточного блоку. Геш також називають сполучним, оскільки для його обчислення береться адреса попереднього блоку;
4) інформація – дані, які передаються користувачем (для прикладу голос за певного кандидата і т.д);
5) список попередніх транзакцій.

На рисунку 1.1 зображена схема роботи технології блокчейн при обробці транзакції:

Рисунок 1.2 – Схема роботи технології Blockchain

Процес обробки транзакції в Blockchain передбачає такі 5 етапів:

1) Запит на здійснення транзакції. Користувач, який хоче надіслати дані іншому користувачеві, формує транзакцію та відправляє його до мережі. Кожен блок блокчейну зберігає дані та геш попереднього блоку. Геш - це унікальний математичний код, який залежить від даних блоку. Система створює унікальний ключ для доступу до надісланих даних. Відправник передає цей ключ одержувачу.
2) Створення транзакції та нового блоку. Нові транзакції обробляються системою і вона утворюює блок, що містить зашифровану інформацію від інших користувачів. Якщо інформація в блоці буде модифікована, це означає що геш блоку також модифікується. З'єднання блоків за допомогою унікального геш-ключа - це те, що робить блокчейн безпечним і захищеним від додавання несправжніх блоків.
3) Поширення нового блоку серед усіх вузлів. Система розміщена у всіх користувачів одночасно, і кожен раз перевіряється, чи відповідає копія інформації, яка була раніше додана в базу. Нові дані будуть одночасно передані у всі екземпляри бази даних для перевірки.

4) Приєднання нового блоку до всіх копій блокчейну. Якщо під час перевірки блок буде визнаний належним, його буде додано в усі копії та приєднано до існуючого ланцюга. Система забезпечить цифровий підпис, за допомогою якого новий блок може бути ідентифікований. Якщо під час перевірки нового блоку система визнає його невідповідним, то він не буде додаватися до інших копій, тому транзакція не буде здійснена.
5) Завершення транзакції. Після створення нового блоку, отримувач може користується унікальним ключем, який був переданий відправником, для отримання відправленої інформації.

В основі технології Blockchain лежить доволі простий і логічний принцип роботи, але при цьому він складно реалізується в житі. Оскільки для створення бази, що розподіляється між усіма вузлами системи одночасно, потрібні відносно великі виробничі потужності та ефективні криптографічні алгоритми.

Модель управління блокчейном може бути класифікована у двох вимірах: “дозволено / без дозволу” та “державний / приватний” [30]. Перший вимір відноситься до

можливість брати участь у механізмі консенсусу
тоді як другий пов'язаний з можливістю для а
користувачеві отримати доступ до належного додатку блокчейну. Більше
зокрема:
• У бездозвольних блокчейнах будь-хто, в тому числі
злісні актори, можуть брати участь у консенсусі
процес. Таким чином, кожен може вільно брати активну участь у
мережі. Витрати вищі, а швидкість повільніша
ніж на дозволеному ланцюжку;
• Дозволені блокчейни зберігаються централізовано
один (або більше) авторизованих користувачів. У цьому випадку
уповноважені користувачі перевіряють кожну транзакцію. Читайте
дозволи можуть бути загальнодоступними або обмежені
довільний обсяг.
З іншого боку, як дозволені, так і
без дозволу блокчейн може бути як загальнодоступним, так і
приватний.

Блокчейн є видом розподіленого збереження даних, що ґрунтується на трьох технологіях: база даних, шифрування та однорангові мережі. База даних представляється у вигляді ланцюга блоків, які певним чином зашифровані та зберігаються в однаковій формі на всіх вузлах мережі.

Для виконання транзакцій у мережі блокчейн, ця мережа повинна містити активні вузли, які можуть затверджувати транзакції. Вузли називають обчислювальними, оскільки вони виконують необхідну роботу для обробки і перевірки транзакцій. Блоки пов’язуються алгоритмами захисту інформації, а саме функціями гешування, тому дані

надійно збережені і не можуть бути модифіковані [5. С. 223]. Кожен блок ланцюга містить зашифровані дані разом з гешом попереднього блоку.

Транзакція вважається корректною тоді, коли геш попереднього блоку корректний. Якщо хтось з-зовні спробує змінити дані в середині блоку, геш, що міститься наступному блоці також зміниться і буде виявлене порушення. Оскільки змінений геш не буде рівний оригінальному. Такий механізм гарантує незмінність ланцюга блокчейну, так як зміни, внесені до інформації блоків, відображаються по всьому ланцюгу мережі та легко будуть виявленими [5. С. 225].

1.1.3 Основні переваги технології

Технології Blockchain властиві наступні особливості:

- Децентралізованість. Системи, що будуються на основі технології Blockchain за своєю природою є децентралізованими, це означає, що ні один користувач чи група користувачів не є права власника всієї мережі. У той час як кожен в мережі має копію розподіленої бази даних, ніхто не може змінити її структуру самостійно. Ця унікальна особливість технології Blockchain забезпечує прозорість і захищеність, до того ж даючи владу користувачам.
- Одноранговий тип мережі (peer-to-peer network). При використанні технології Blockchain, взаємодія між двома користувачами на основі однорангової моделі виконується без небхідності участі третьої сторони [7]. Протокол peer-to-peer, на якому базується технологія блокчейн, дозволяє всім користувачам мережі зберігати однакову копію блоків з транзакціями, що затверджується за допомогою алгоритму досягнення консенсусу. Для прикладу, якщо потрібно здійснити якусь транзакцію з одного вузла мережі в інший, які можуть бути фізично далеко один від одного, це може бути виконано за допомогою технології Blockchain самостійно майже за декілька секунд. Також будь-які перебої чи додаткові витрати не будуть вираховуватись при передачі.
- Незмінність. Особливість незмінності блокчейну відноситься до того, що дані, які були записанні в блок ланцюга, не можуть бути відредаговані

ніяким чином. Для зрозуміння цієї особливості, можна розглянути як приклад, відправка електронної пошти. Після того, як лист надіслано людям, його не можна повернути назад. Щоб обійти цю особливість, можна попросити всіх отримувачів видалити надісланий електронний лист, а це не завжди працює [8]. Після того як транзакція була оброблені, дані блоку неможливо видозмінювати чи перероблювати. У разі, якщо необхідно буде модифікувати дані одного блоку, доведеться модифікувати увесь наступний ланцюг після цього блоку, так як кожен блок містить в собі геш попереднього блоку. А зміна в одному геш-значенні приводить до модифікацій у всіх гешах після цього блоку. Це дуже складно оновити всі геші, так як процес генерування нового гешу вимагає досить великої обчислювальної потужності [9]. Таким чином, дані, які зберігаються в блокчейні, не сприйнятливі до модифікацій чи несанкціонованого втручання через властивість незмінності.

- Захищений від зовнішнього втручання. Внаслідок особливості незмінності, фальсифікація даних у блоках ланцюга без модифікації інших блоків стає неможливою. Блокчейн запобігає зовнішньому втручанню, через те що будь-які зміни хоча б в одному блоці можуть бути легко виявленими і вирішені. Визначають два ключових способи виявлення фальсифікацій: перевірка геш-значень і блоків [10].

1.1.4 Використання технології для процедури голосування

Компанії працюють разом з метою обміну послугами чи над спільними продуктами. Всі умови надання та прийняття послу підписуються зацікавленими сторонами у формі угод чи контрактів. Однак ці паперові контракти схильні до помилок та шахрайства, що кидає виклик фактору довіри між сторонами та підвищує ризики. Блокчейн пропонує надзвичайне рішення цієї проблеми за допомогою Розумних Контрактів [11].

Розумні контракти виконують аналогічні функції, як угоди на паперовій основі. Відмінним фактором щодо розумних контрактів є те, що вони є цифровими, а також самореалізованими за своєю природою. Самореалізація означає, що коли певні умови в кодексі цих договорів виконуються, вони автоматично вступають в силу. Ethereum, платформа з відкритим кодом блокчейн представила розумні контракти в екосистемі Blockchain. Розумні контракти можна використовувати для різних ситуацій чи галузей, таких як фінансові угоди, медичне страхування, документи на майно нерухомості, краудфандинг тощо.

Наприклад, смарт-контракти Blockchain можуть використовуватися в охороні здоров'я для управління постачанням рідкісних препаратів.

Після того, як певна кількість лікарського засобу буде відвантажена виробничою компанією для доставки фармацевту, заздалегідь, може бути створений розумний контракт з усіма даними, такими як інформація про препарат, кількість поставок тощо. Цей розумний контракт несе відповідальність за постачання по всьому ланцюгу поставок між різними посередниками. Оскільки розумний контракт працює на певних визначених умовах, ніхто не може їх змінити чи внести будь-які зміни в договір, забезпечивши довіру та справжність ліків.Кожен користувач має пару приватного та відкритого ключів.

Використовується приватний ключ, який зберігатиметься в таємниці для підписання транзакцій. Цифрові підписані транзакції транслюється по всій мережі. Типовий цифровий підпис бере участь у двох етапах: фаза підписання та етап перевірки. Наприклад, користувач Аліса хоче надіслати інший користувач Боб повідомлення. (1) На етапі підписання, Аліса шифрує свої дані своїм приватним ключем і відправляє Бобу зашифрований результат та вихідні дані. (2) На етапі верифікації Боб перевіряє значення за допомогою відкритого ключа Аліси. Таким чином, Боб міг легко перевірити, чи були дані підроблені чи ні.

Типовим алгоритмом цифрового підпису, який використовується в блокчейнах, є

алгоритм цифрового підпису еліптичної кривої (ECDSA) [11].

Вибори уряду. Незалежно від того, наскільки безпечними є урядові вибори, шанси на шахрайство через антисоціальні елементи завжди зберігаються. Нинішня система голосування покладається на ручну обробку голосів та довіру населення. Навіть, якщо порушення безпеки та шахрайство усунені, шанси помилок вручну не можна ігнорувати. У таких випадках найкращим рішенням є автоматизація загального процесу за допомогою розумних контрактів [12].

Інтелектуальні контракти Blockchain забезпечують сучасну систему, завдяки якій ці поширені проблеми можна легко усунути. Записи в смарт-контрактах дозволять забезпечити прозорість та безпеку, зберігаючи при цьому конфіденційність виборців, забезпечуючи чесні вибори [13].

З кожним днем світ стає все більш оцифрованим. Наприклад, розглянемо фінансові операції, які відбуваються в Інтернеті, ви можете легко увійти за допомогою своїх облікових даних, щоб отримати доступ до своїх коштів. Однак у цьому випадку ніхто не може ідентифікувати особу, яка користується грошима. Якщо ім’я користувача та пароль хтось зламає, немає жодного способу захистити власні кошти.

Необхідною у наш час є система, яка б забезпечувала індивідуальну ідентифікацією в Інтернеті. Технологія розподіленої мережі, що використовується в блокчейн, пропонує передові методи шифрування публічно-приватного користування, за допомогою яких можна довести свою особу та цифрувати свої документи [14]. Ця унікальна захищена особа може зробити безпечним проведення будь-яких фінансових операцій або будь-яких суспільних взаємодій в Інтернеті. Більше того, розрив між різними державними органами та приватними організаціями можна заповнити універсальним рішенням онлайн-ідентичності, яке може надати блокчейн [15].

Блокчейн може зберегти певну частку конфіденційності завдяки
відкритий та закритий ключі. Користувачі здійснюють транзакції зі своїми
приватний та відкритий ключі без реального виявлення особистості.
Однак показано, що блокчейн не може

гарантувати конфіденційність транзакцій, оскільки цінності всіх транзакції та залишки для кожного відкритого ключа є відкритими видно. Крім того, недавнє дослідження [10] показало, що користувальницькі Операції з біткойнами можна пов’язати, щоб розкрити інформацію користувача.

Більше того, Бірюков та ін. [11] представив метод посилання псевдоніми користувачів до IP-адрес, навіть якщо користувачі позаду Переклад мережевих адрес (NAT) або брандмауери. У [11] кожен Клієнта можна однозначно ідентифікувати за набором вузлів, які він з'єднує до. Однак цей набір можна навчитися і використовувати для пошуку походження операції.

1.1.5. CAP-теорема

Відповідно до теореми CAP, розробникам потрібно вибрати три умови в розподіленій системі: узгодженість, доступність та допуск розподілу. Зокрема, теорема говорить нам, що ми можемо досягти лише двох із трьох умов одночасно [16]. Е. Брюер цілком довів цю причину.

Є багато спроб створити систему, яка може одночасно задовольнити всі три умови, але всі вони марні.

Якщо два вузли отримують два суперечливі запити від клієнта, вони повинні прийняти та обробити запит, тим самим порушуючи властивість узгодженості, або принаймні один із них не повинен приймати запит, тим самим порушуючи властивість доступності [17]. .

Таким чином теорема ділить розподілені системи на 3 класи:

1) CA-На перший погляд, такий тип системи здається ідеальним, оскільки він може одночасно забезпечити доступність та цілісність даних. Але насправді така система може існувати лише на одному сервері, тобто її не можна розповсюджувати.

Помітним прикладом системи ЦС є реляційна база даних, яка відповідає вимогам цілісності та доступності даних.

2) AP-Цей тип системи може забезпечити доступність даних у будь-який час, але не може гарантувати цілісність цих даних.

Звичайно, поява таких систем набагато раніше теореми CAP, наприклад DNS, але зростання популярності узгоджується із поширенням принципу (особливо деякі нереляційні бази даних не можуть гарантувати цілісність результатів, але Забезпечте його доступність, посилаючись на теорему).

3) CP- Цей тип системи також є загальним для питань, які відіграють важливу роль у цілісності. Інтегрована система може гарантувати коректність даних на всіх вузлах, але не може гарантувати їх доступність.

Прикладом такої системи є програмне забезпечення розподіленої фінансової системи, де узгодженість даних має найвищий пріоритет, наприклад, мережі банкоматів [18].

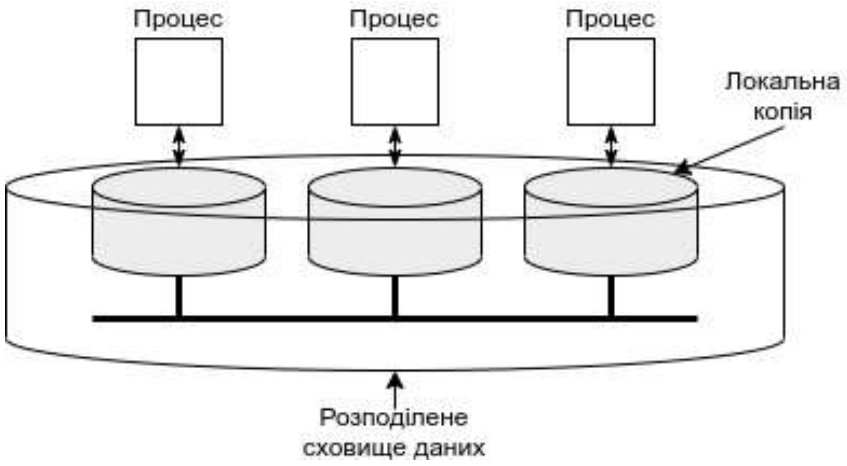

Рисунок 1.2 – Узагальнена організація логічного сховища даних, фізично розподіленого по декількох процесах

12 Огляд існуючих рішень

1.2.1 Сучасні системи електронного голосування

Концепція електронного голосування стає дедалі популярнішою у Європі. У Естонії онлайн-голосування представили у 2005, наразі більшість громадян голосує онлайн, але при цьому для альтернативи зберегли традиційну систему. Оскільки естонська система є прикладом системи, якої хотілося б досягти, тому розглянуто саме її.

Ця система базується на національному посвідченні особи, яке видається всім жителям країни. Ці посвідчення видаються у формі карти, що містить зашифровані дані, які використовуються для ідентифікації власника. Картка дозволяє здійснювати низку електронних видів діяльності, включаючи здійснення цифрового підпису для документів, онлайн-банкінг, доступ до інформації громадянина в державній базі даних та онлайн-голосування.

Щоб мати змогу проголосувати, виборець прикладає свою картку до спеціального зчитувача карток, і здійснює авторизацію на сайті з голосуванням. Далі виборець вводить свій PIN-код і система перевіряє, чи має він право на голосування. Після здійснення голосування виборець може редагувати свій голос впродовж двох днів до дня закінчення виборів. Виборець також має можливість ідентифікувати себе за допомогою мобільного телефону. Однак для такого процесу необхідні спеціальна SIM-карта для мобільного.

Коли користувач віддає свій голос, цей голос проходить через загальний сервер перенаправлення, який зашифровується і зберігається до закінчення виборів. Після закінчення виборів голоси записуються на DVD-диски, видаляються із сервера, а потім передаються на сервер для підрахунку. Цей сервер є відключеним від усіх мереж. Він розшифровує та рахує голоси, а потім відображає результати. Всі етапи процесу голосування є зареєстрованими та перевіреними.

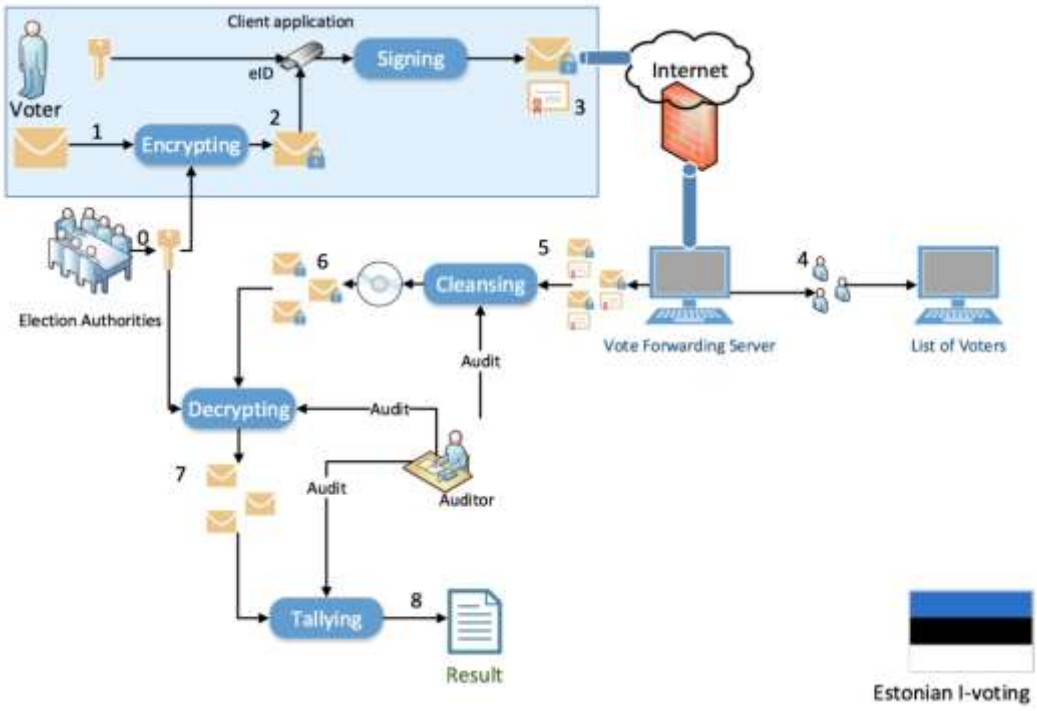

Рисунок 1.2 – Естонська система електронного голосування [21]

На місцевих виборах 2013 року дослідники вивчали таку систему голосування та виявили багато можливих вразливостей. Однією з таких вразливостей є здатність встановити на комп'ютері на стороні клієнта шкідливого програмного забезпечення, яке відслідковує дії виборця, і після здійснення вибору, без відома виборця, змінює його голос на користь іншого кандидата. Інша вразливість полягає у тому, що зловмисник має можливість без ніяких проблем інфікувати DVD-диски, що використовуються для конфігурування серверів і перенесення голосів виборців [22].

1.2.2 Огляд існуючих реалізацій

Розглянемо наступні приклади існуючих систем онлайн-голосування:

- **Agora.**

В даний час єдина система, що використовує технологію Blockchain для онлайн-голосування. Після авторизації виборець може створювати нові голосування та брати участь у тих виборах, до яких він має доступ.

Переваги:

1) нескладний та інтуїтивно зрозумілий інтерфейс;
2) зручний процес авторизації;
3) надійність та безпека даних процесу голосування.

Недоліки:

1) позиціонування системи як додаток для недержавних голосувань;
2) авторизація не є надійною;

- **Polys.**

Система працює зі звичайною базою даних, але при цьому всі дані зберігаються у мережі блокчейн. Після авторизації та підписки на оплати, користувач може створити нове голосування, та брати участь в доступних голосуваннях.

Переваги:

1) нескладний та інтуїтивно зрозумілий інтерфейс;

Недоліки:

1) оперування звичайною базою даних, а саме дані виборців збережені в реляційній базі даних;
2) висока ціна продукту.

ВИСНОВКИ ДО РОЗДІЛУ 1

У цьому розділі розглядаються існуючі впровадження систем електронного голосування, які забезпечують надійність та безпеку результатів голосування. Досліджено існуючі найпопулярніші рішення таких систем. Кожен із них має низку переваг і недоліків. За домогою аналізу можна визначити оптимальні вимоги до функцій системи для задоволення основних потреб та досягнення максимальної простоти для використання.

На даний момент жодна країна не використовує технологію Blockchain для електронного голосування. Хоча ця технологія має великий потенціал і у майбутньому можливо буде використана.

Основними недоліками традиційних системи онлайн-голосування є:

1) централізація результатів голосування;
2) вразливість до хакерських атак;
3) прив’язка до документів конкретної країни, що робить неможливим використання системи в Україні.

Розглянемо основні переваги використання технології блокчейн для електронного голосування:

1) децентралізація;
2) висока надійність та безпека;
3) захищеність результатів
4) немінюваність.

Розглянемо основні вимоги до функцій системи:

1) використання паспорту громадянина України;
2) захищеність даних як користувача,голосування;
3) простота використання;
4) інтуїтивно зрозумілий інтерфейс.

Тому, щоб усунути ці недоліки, потрібно розробити додатки із зазначеними вище вимогами. Для того, щоб система могла вирішувати конфлікти даних, може бути використаний алгоритм перевірки роботи або алгоритм для вирішення проблеми консенсусу машини для обробки нових блоків. Щоб скоротити час розробки, рекомендується використовувати мову програмування C ++.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ДОДАТКУ

2.1 Опис предметної області

Предметна область — частина реального світу, яка підлягає вивченню з метою організації управління, а потім автоматизації. Предметна область представляється множиною фрагментів. Кожен фрагмент предметної області характеризується безліччю об’єктів і процесів, що використовують об’єкти, а також безліччю користувачів, які характеризуються різними поглядами на предметну область. Предметну область прийнято розглядати у вигляді трьох представлень:

1) представлення предметної області в тому вигляді, в якому вона дійсно існує;

2) як її сприймає людина (мається на увазі проектувальник бази даних);

3) як вона може бути описаною за допомогою символів.

Тобто проектувальник бази даних оперує такими поняттями як реальність, опис (подання) реальності і дані, які відображають це представлення.

Опис предметної області представляється у вигляді трьохрівневої схеми (її також називають модель ANSI/SPARC). Схема зображена на рис. 2.1:

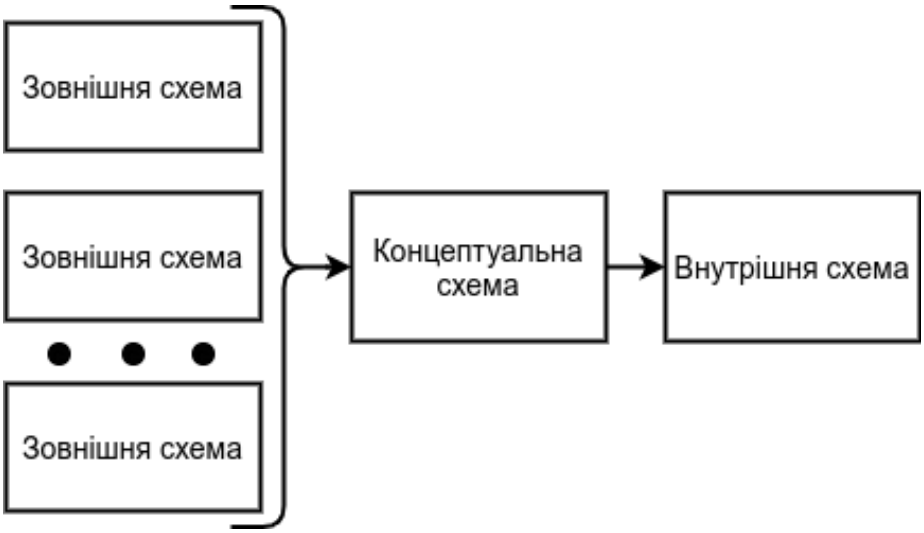

Рисунок 2.1 – Схема предметної області

Предметну область поділяють на три представлення (тобто схемами). Зовнішнє представлення (зовнішня схема) даних є сукупністю вимог до даних зі сторони деякої конкретної функції, що виконує користувач. Концептуальна схема є повною сукупністю всіх вимог до даних, отриманої з користувацьких представлень про реальний світ. Внутрішня схема - це сама база даних [23].

Звідси випливають основні етапи, на які розбивається процес проектування бази даних системи:

1) концептуальне проектування;

2) логічне проектування;

3) фізичне проектування.

Концептуальне проектування — збір, аналіз і редагування вимог до даних.

Для цього здійснюються наступні заходи:

- обстеження предметної області, вивчення її інформаційної структури;
- визначення всіх фрагментів, кожен з яких характеризується користувацьким представленням, інформаційними об'єктами і їх зв'язками, процесами над інформаційними об'єктами;
- моделювання і об’єднання всіх представлень (інтерфейсів).

Після закінчення даного етапу отримуємо концептуальну модель, інваріантну до структури бази даних. Часто вона представляється у вигляді моделі “сутність-зв'язок”.

Логічне проектування — це перетворення вимог до даних в структури даних. В результаті одержується структура бази даних, що орієнтована на блокчейн технологію, та специфікацію прикладної програми.

Фізичне проектування — визначення особливостей методів зберігання даних, доступу до даних, і т.д.

На рисунку 2.2 зображена схема структури рівнів проектування та представлень:

Рисунок 2.2 – Схема структури рівнів проектування та представлень

В контексті даної магістерської дисертації система онлйн-голосування управляє мережею Blockchain. Кожен новий голос користувача додається у вигляді блоку до ланцюгу блокчейн мережі. Користувач відправляє запити до блокчейн системи, і система послідовно записує необхідну інформацію, шифрує та додає її до новоствореного блоку. Алгоритм досягнення консенсусу використовується для запобігання конфліктам записів та перевірки блоків ланцюга блокчейну [24].

Досягнення консенсусу здійснюється на основу голосування. Консенсус – це концепція прийняття рішення, яка задовольнить більшість респондентів.

Існує два дійсних випадки несправності функціонування розподілених систем:

1) Збій вузла. Проблема полягає в тому, що несправний вузол стає недоступним для решти системи, тому якщо вузол містив унікальні дані, система втратить цілісність. Найпоширенішими прикладами такої ситуації є: збій сервера або будь-якого з його компонентів, збій системи збереження даних, збій операційної системи, втрата мережевого з'єднання, тощо;

2) Візантійська помилка. Ця проблема цікава тим, оскільки вузол продовжує працювати, але працює неправильно. Цей випадок несправності найважче виявити та виправляти. Наприклад, пошкодження пакетів тощо.

Онлайн-голосування забезпечує такі основні переваги:

- Підвищений рівень участі. Системи онлайн-голосування, як правило, максимізують участь користувачів, дозволяючи їм проголосувати будь-де і забезпечують доступ до голосування з будь-яких комп'ютерних мереж та підключених до інтернету пристроїв.
- Безпека. Коли говориться про про безпечні вибори, мається на увазі забезпечення рівнів безпеки, які реалізуються таким чином, щоб підраховані голоси відповідали волевиявленню виборців і були проголосовані виборцями, яким надано доступ до участі у виборах. В процесі онлайн-голосування, крім логічного і фізичного рівнів захисту, створюються деякі механізми, що надають доступ лише тим користувачам, уповноваженими офіційними документами, забезпечуючи тим самим усі гарантії безпеки, що роблять процес онлайн-голосування та традиційний виборчий процес еквівалентними [26].
- Доступність. Онлайн-голосування забезпечує захищений приватний канал, завдяки якому всі користувачі можуть брати участь на рівних умовах. Громадяни країни, що живуть закордоном, а також громадяни, які мають обмеженість в мобільності або перебувають у подорожжі, можуть проголосувати віддалено. Тому це позитивно впливає на показник явки на виборах, а, потім, і на законність виборів.

- Прийнятність. За домогою технології онлайн-голосування можна завершити весь процес голосування до кінця. Дизайн системи дозволяє адміністраторам гарантувати, що користувачі можуть правильно проголосувати та їхні голоси підраховані відповідно до вибору їхнього варіанту. Крім того, кожному користувачу видається квитанція про голосування.
- Ефективність. Порівнюючи з традиційним голосуванням на папері, зменшення витрат на організацію і здійснення голосування може значно підвищити ефективність управління процесом виборів.
- Точність. Онлайн-голосування усуває помилки підрахунку вручну, в результаті чого отримуються точні та швидкі оголошення результату, і кожен голос може бути перевірений користувачем, який хоче отримати підтвердження.
- Надійність. Алгоритм криптографічного захисту є основою для надійності та конфіденційності процесу голосування. Протокол шифрування забезпечує анонімність виборців, цілісність голосів, проведення аудиту виборів. Аудит виборів безпечніший, аніж у традиційному голосуванні на папері, де виборець не має контролю над своїм голосом.

2.2. Визначення функцій та вимог

Система має забезпечувати наступні основні функції:

1) надання виборцю можливість реєстрації в системі;
2) можливість створення нового голосування;
3) можливість вибору варіанта голосування;
4) можливість аудиту голосування, а саме перегляд результатів голосування, статистику.

Система онлайн-голосування має такі вимоги:

1) можливість зручної та зрозумілої авторизації і голосування;
2) інтуїтивний процес голосування при цьоому дотримуючись усім вимогам;
3) простота у використанні системи для охоплення нових потенційних користувачів.

2.3. Опис функціоналу системи

Система повинна надавати користувачеві наступний функціонал:

1) реєстрація в системі;
2) перегляд усіх кандидатів виборів;
3) віддання голосу користувача за обранного кандидата;
4) надсилання сповіщення для користувача про вдале голосування.
5) перегляд статистики системи голосування.

2.4. Прецеденти

Визначивши функції, які система повинна забезпечувати, можна детально розглянути сценарії користування програмою і побудувати діаграму варіантів використання. Діаграма прецедентів системи зображена на рис. 2.2. Діаграма показує можливі дії користувача, які є абстрагованими від деталей.

Рисунок 2.2 – Діаграма прецедентів додатку

Можна детальніше розглянути прецеденти:

1) реєстрація в системі;
2) віддання голосу;
3) отримання сповіщення.

2.4.1 Реєстрація в системі

Реєстрація в системі це один з основних прецедентів у системі і виконується під час запуску програми. При реєстрації буде створений захищений обліковий запис користувача, який теж є частиною мережі блокчейн (блок ланцюга). Таким чином, усі особисті дані користувача будуть надійно захищені та зашифровані. На рис. 2.3 зображено ієрархію даного прецеденту.

Рисунок 2.3 – Ієрархія прецеденту «Реєстрація в системі»

Можливі два варіанти реєстрації користувача в системі:

1) реєстрація нового облікового запису (таблиця 2.1);
2) авторизація в існуючий обліковий запис (таблиця 2.2);

Таблиця 2.1 – Прецедент «Реєстрація нового облікового запису»

Користувач	Система
1. Введення особистих даних і пароль у спеціальне вікно реєстрації системи	2. Створення нової транзакції, шифрування вхідних даних, формування блок і приєднання його до ланцюга
	3. Присвоєння користувачу персональний цифровий ключ

Таблиця 2.2 – Прецедент «Авторизація в існуючий обліковий запис»

Користувач	Система

1. Введення персонального цифрового ключа та паролю в спеціальному вікно авторизації в системі	2. Перевірка наявних голосування для користувача
	3. Надання доступу до відкритих голосувань

2.4.2 Віддання голосу за певного кандидата

Система повинна забезпечувати анонімність і захист від несанкціонованого втручання процесу віддання голосу. На рис. 2.4 зображено ієрархію прецеденту перегляду голосування. Кожен користувач отримує унікальний цифровий ключ при реєстрації (таблиця 2.3). Такий ключ є схожим до цифрового підпису.

Рисунок 2.4 – Ієрархія прецеденту «Перегляд відповідного голосування»

На рис. 2.5 показана ієрархія прецеденту віддання голосу за певного кандидата.

Рисунок 2.5 – Ієрархія прецеденту «Віддання голосу»

Таблиця 2.3 – Опис прецеденту «Віддання голосу»

Користувач	Системи
1) Віддання голос за певного кандидата та підтвердження вибору своїм	2) Збереження даних про вибір користувача, шифрування цих даних, створення нової транзакції та блоку, що буде

персональним цифровим ключем	приєднаний до ланцюга блокчейну

2.4.3 Отримання сповіщення

Прецедент отримання сповіщення виникає у випадку, коли користувач зробив вибір бажанного кандидата і завершив роботу з додатком. Статистика роботи системи та результати виборів подаються у форматі html сторінки, для зручного перегляду.

Таблиця 2.4 – Опис прецеденту «Отримання сповіщення»

Користувач	Система
1) Віддання голосу та завершення роботи з системою	2) Відображення вікна в якому показана інформація про успішно зарахований голос
	3) Відправлення електронного листа на пошту користувача з підтвердженням голосування
	4) Завершення поточного голосування для даного користувача

2.5 Діаграма класів моделі даних виборця

Використовуючи мову програмування C ++, ви можете використовувати об’єктно-орієнтоване проектування для створення систем.

Беручи до уваги дані, витягнуті з розглянутих раніше прецедентів, може бути побудована діаграма класу структури даних моделі програми. Мова програмування C ++ використовується при розробці алгоритму розподілу запитів у додатку. Об'єктно-орієнтовані методи та шаблони проектування використовуються для моделювання роботи мережі Blokchain. Це дозволяє імітувати роботу мережі блокчейнів. Для моделювання роботи алгоритму для досягнення консенсусу використовується мова програмування C ++. Бібліотека std :: hash використовується для створення хеш-значень. Інтерфейс користувача візуальної частини програми розроблений з використанням мови

програмування C ++ у поєднанні з фреймворком Qt та декларативною мовою QML. На малюнку. 2.6 показана діаграма класів моделі даних користувача.

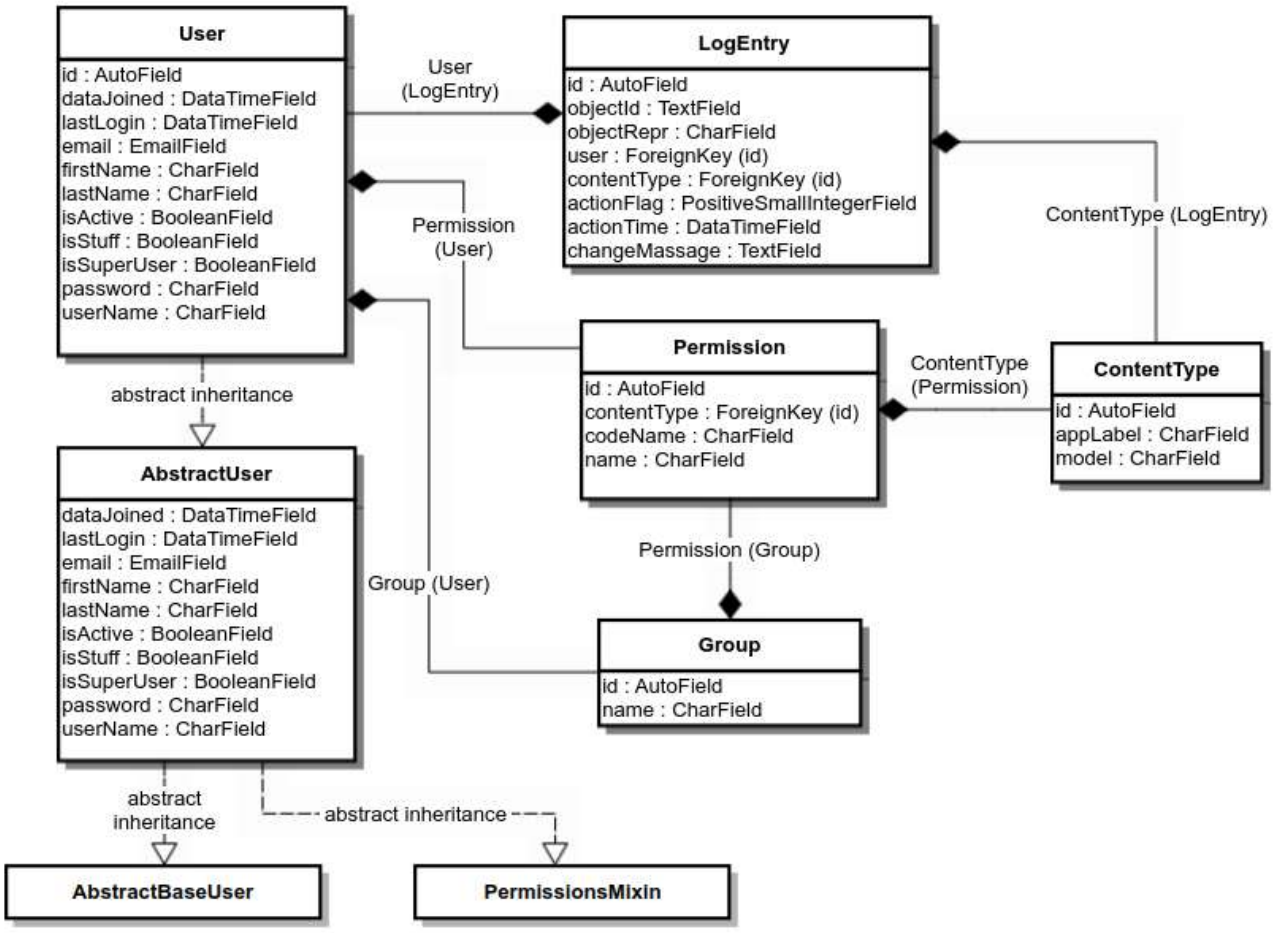

Рисунок 2.6 – Діаграма класів моделі даних виборця

2.6 Алгоритм досягнення консенсусу

Як вже згадувалося вище, проект використовує консенсус для побудови алгоритмів - доказ роботи доречний. На малюнку. 2.7 показана загальна схема роботи алгоритму.

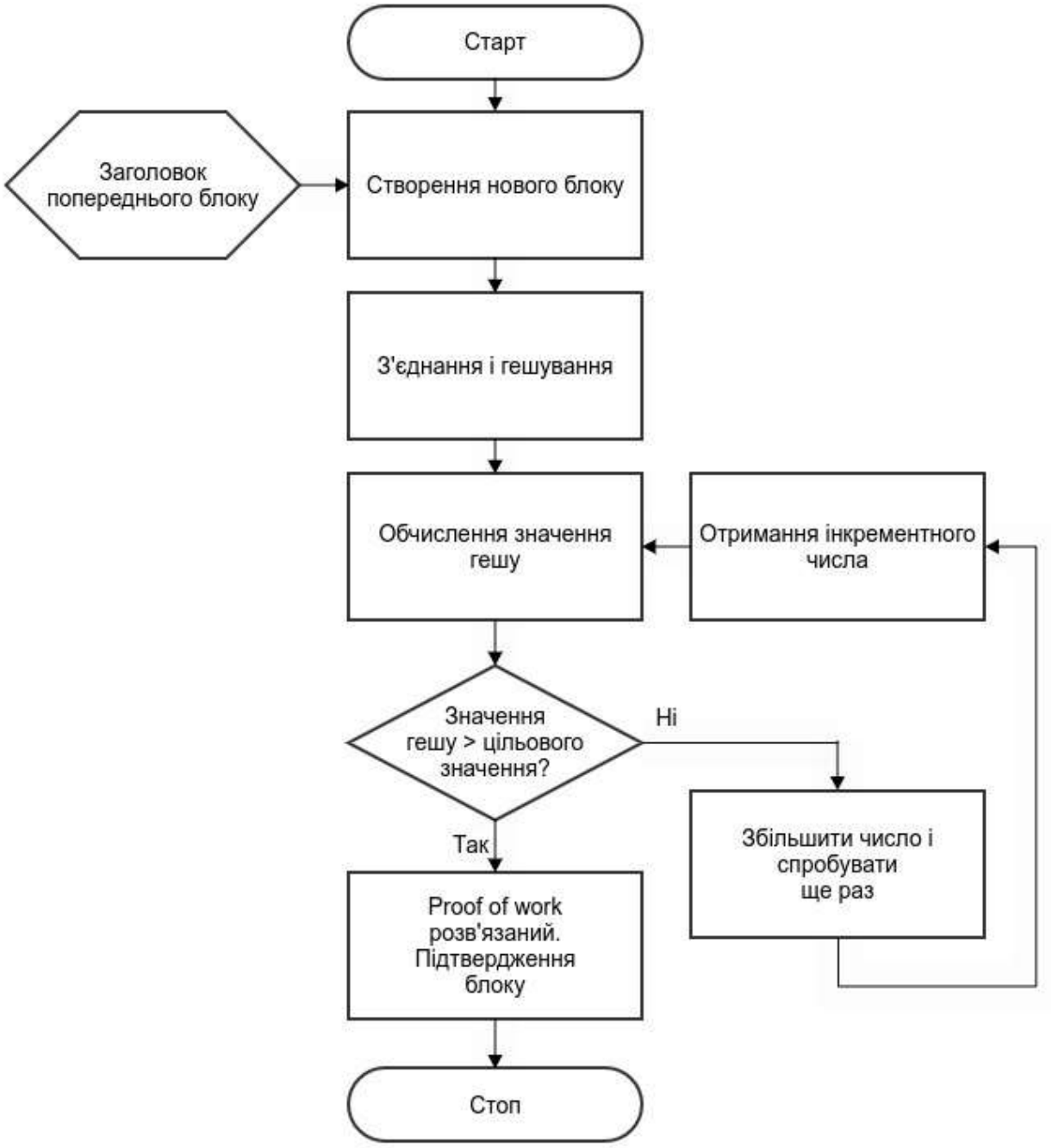

Рисунок 2.7 – Схема роботи алгоритму Proof of work

Розглянемо алгоритм більш детально, опишемо його робочі стадії та розглянемо особливі випадки.

2.6.1 Proof of work – алгоритм досягнення консенсусу

Proof of work - це алгоритм досягнення консенсусу з використанням журналу, про який буде написано далі.

Доказ роботи досягається консенсусом, тобто спочатку вибирається менеджер, а менеджер несе повну відповідальність за управління журналом. Лідер отримує запит від клієнта, переадресовує його на інші сервери та повідомляє сервер про безпечний вхід. Наявність лідера спрощує управління журналом реплікації. Наприклад, лідер може вирішити, куди записувати нові записи, не обмінюючись повідомленнями з іншими серверами, а потім передавати прості дані від лідера іншим серверам. Якщо керівник не вдається або втрачає контакт, буде обраний новий керівник [27. С. 6].

Використовуючи метод керівництва, доказ робочого навантаження ділить проблему консенсусу на три відносно незалежні підзадачі, які будуть розглянуті в наступних розділах:

• Вибори лідера: Коли нинішній лідер відмовляється, повинен бути обраний новий лідер.

• Тиражування щоденника: керівник повинен приймати записи щоденників від клієнтів і відтворювати їх протягом кластера, змушуючи інші щоденники погоджуватися.

• Безпека: Ключовим атрибутом безпеки доказу роботи є атрибут безпеки стану комп’ютера. Якщо який-небудь сервер застосував запис журналу на своєму комп’ютері, інший сервер не зможе застосувати різні команди до того самого індексу журналу. [27. С. 7].

Кластер перевірки роботи повинен складатися з непарної кількості серверів. Сервер може бути в одному з трьох станів: кандидат, послідовник або лідер. За звичайних обставин лідер повинен бути лише один, а всі інші сервери повинні бути імітаторами. Послідовники пасивні. Вони не можуть створювати або подавати будь-які запити самостійно. Вони реагуватимуть на запити кандидатів та керівників. Усі запити від клієнта спеціально обробляються лідером. Якщо клієнт

відправляє запит і досягає послідовника, послідовник пересилає запит лідеру, а потім лідер обробляє його. Коли зв'язок з лідером від'єднаний, сервер змінює статус імітатора на кандидата, і кандидат, що набрав найбільшу кількість голосів, стане новим лідером [27. С. 11].

Доказ роботи t ділить час на кілька термінів. Цей термін є нумерованим послідовним цілим числом. Кожен новий термін починається з обрання нового керівника. Якщо кандидат виграє вибори, він виступає як довгостроковий лідер. У випадку, коли жоден кандидат не набере більшість голосів, тобто голоси розподіляються рівномірно, термін повноважень закінчується без керівника та нового терміну, і починається нове голосування.

2.6.2. Основні стани

Як зазначалося вище, сервери кластера мають три основні стани. Будь-який сервер одночасно може знаходитись лише в одному стані. Перехід з одного стану в інший залежить від ситуації в кластері.

Більше того, особливістю цього алгоритму є використання періодів часу для лінеаризації часу. Отже, алгоритм вирішує проблему синхронізації між серверами, які потенційно можуть бути в різних часових поясах.

Оскільки ми не можемо припустити, що існує глобальний синхронний годинник, вартість товару зростає монотонно, реалізуючи таким чином загальний порядок подій. Кожен сервер зберігає своє бачення терміну в постійному сховищі. Оновлюйте лише тоді, коли серверний термін перезапускає вибори або дізнається від іншого сервера, що його термін застарів. Всі повідомлення містять елемент вузла, який перевіряється сервером-одержувачем. Якщо часу джерела мало, відповідь - ні. Якщо термін отримання одержувача коротший, оновіть його термін перед аналізом повідомлення, щоб їхні терміни були однаковими [28].

У наступних параграфах буде детально обговорено процес переходу з одного стану в інший.

2.6.3. Процедура вибору лідера

Лідером є активний сервер, який приймає запити від клієнтів і розсилає їх усім послідовникам. Як тільки новий вузол підключається до кластера, він стає імітатором і починає отримувати запити від лідера або кандидата. Керівник регулярно надсилає імпульси всім передплатникам, щоб повідомити їх про своє існування та діяльність.

Коли лідер відмовляється або втрачає з ним контакт, майбутній кандидат оголосить себе кандидатом і розпочне вибори, продовжуючи тим самим свій термін. Кожен сервер у державі-кандидаті завжди голосує за нього [29].

Кандидат перебуватиме у такому стані, поки не відбудеться одна з наступних подій::

3) він переможе на виборах;
4) Лідером буде інший вузол, який розпочне вибори раніше і набере більше голосів;
5) Термін пройшов без чіткого лідера.

Далі розглянемо кожну подію більш докладно.

2.6.4. Кандидат перемагає

Якщо він отримає голоси від більшості серверів кластеру протягом того самого періоду часу, кандидат виграє вибори. Кожен сервер буде голосувати не більше ніж за одного кандидата протягом визначеного періоду часу за принципом "хто прийшов"[30]. Правило більшості гарантує, що принаймні один кандидат може перемогти на певних виборах. Поки кандидат виграє вибори, він стає лідером. Потім він надсилає імпульс повідомлення на всі інші сервери для встановлення своєї діяльності та запобігання новим виборам [31].

2.6.5. Перемога іншого кандидата

Під час очікування голосування кандидат може отримати RPC AppendEntries з іншого сервера, який претендує на роль лідера. Якщо термін лідера є принаймні таким же, як поточний термін кандидата, тоді кандидат визнає лідера законним і повертається в стан наслідувальника. Якщо термін в RPC менший за поточний термін кандидата, кандидат відмовляється від RPC і продовжує в перебувати у стані кандидата [32].

2.6.6. Нічия

Третій можливий результат полягає в тому, що кандидат не виграє і не програє вибори: якщо декілька послідовників стають кандидатами одночасно, голоси можуть бути розділені таким чином, що жоден кандидат не отримає більшість. Коли це трапляється, кожен кандидат припиняє термін і розпочинає нові вибори, збільшивши термін і починаючи черговий раунд запитів-голосування по RPC. Якщо уявити цю ситуацію то можна подумати, що в такому стані кластер може простоювати нескінченну кількість часу, або іншими словами – зависнути. Але розробники алгоритму врахували таку можливість і додали механізм який виводить систему з такого стану [33, 34].

Термін закінчення голосування є випадковим, це дозволяє гарантувати, що рівномірний розподіл голосів є рідкістю. Для запобігання рівномірного розподілу голосів, тайм-аут виборів вибирається випадковим чином з зафіксованого часового інтервалу, найчастіше цей інтервал робиться дуже малим. Це розділяє сервери у часі так, що в більшості випадків лише один сервер досягне тайм-ауту; він переможе на виборах і відправить імпульс, перш ніж будь-який інший сервер [35]. Цей же механізм використовується для обробки роздільних голосів. Кожен кандидат відновлює рандомізований тайм-аут виборів на початку виборів, і чекає, що цей тайм-аут пройде перед початком наступних виборів; це зменшує ймовірність ще одного розколу на нових виборах.

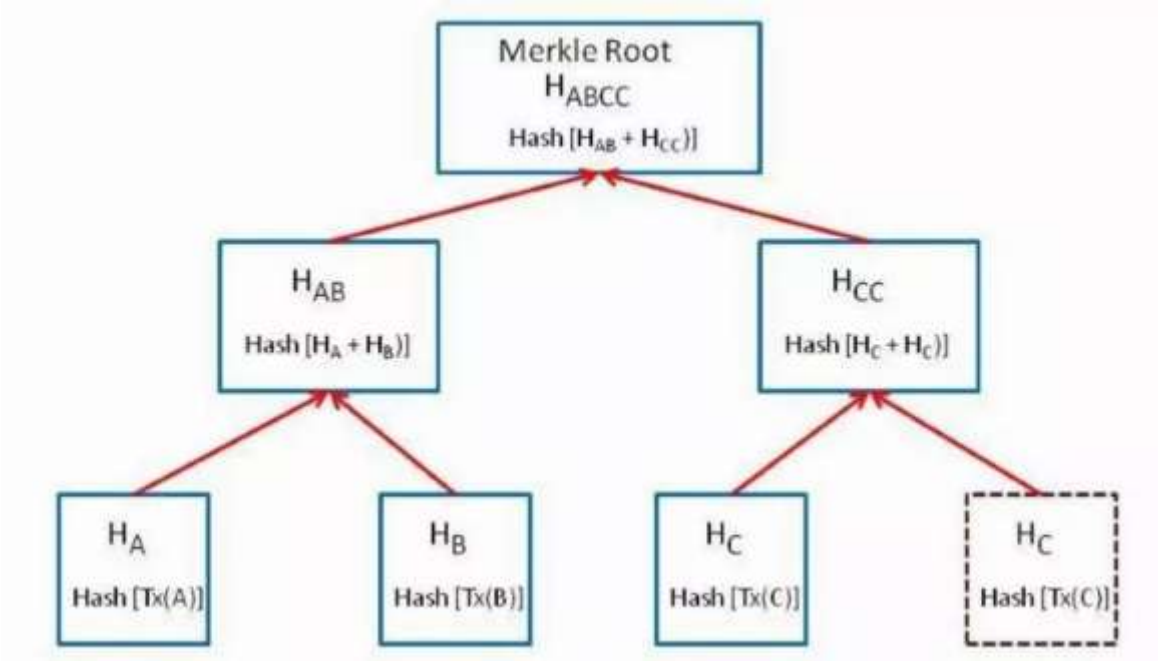

Рисунок 2.8 – Схема формування кореня Меркла

Дерево будується наступним чином:

* Обчислюються хеші транзакцій, розміщених у блоці: хеш (L1), хеш (L2), хеш (L3) тощо.

* Хеші обчислюються із суми хешів транзакцій, наприклад, хеш (хеш (L1) + хеш (L2)). Оскільки дерево Меркла є двійковим, кількість елементів на кожній ітерації має бути парною. Отже, якщо блок містить непарну кількість транзакцій, то остання дублюється і додається до себе: хеш (хеш (L3) + хеш (L3)).

* Далі хеші обчислюються із суми хешів. Процес повторюють, поки не буде отриманий єдиний хеш - корінь меркле. Це криптографічний доказ цілісності блоку (тобто, що всі транзакції відбуваються в зазначеному порядку). Кореневе значення фіксується в заголовку блоку.

Цей кореневий хеш називається корінь Меркла, і завдяки зв’язку хешів у дереві він містить всю інформацію про кожен хеш транзакцій, який існує у блоці. Він пропонує однопозиційне хеш-значення, яке дозволяє перевірити все, що коли-небудь було присутнє в цьому блоці [50].

Наприклад, якщо вам доведеться перевірити транзакцію, яка стверджується, що відбулася з блоку № 137, потрібно лише перевірити дерево Меркла цього блоку, не перевіряючи що-небудь на будь-яких інших сусідніх блоках блокчейна.

По суті, дерево Меркла та корінь Меркла значно знижують кількість хешування, що дозволяє виконувати більш швидкі перевірки та транзакції.

2.8. Розробка підходу для моделювання блокчейн мережі

Кожен блок розробки проекту проекту з трьома основними компонентами:

a. Клієнт;
b. Майнери;
c. Мережа Блокчейн.

2.8.1. Структура блокчейн

На думку IBM, блокчейн - це спільнота, розподілена мережа, яка полегшує записи транзакцій та відстеження активів у мережі. Діяльністю можуть бути матеріальні активи (наприклад, майно, будинки, транспортні засоби) або нематеріальні активи (наприклад, цифрова валюта) та інтелектуальна власність. В основному блокчейн зберігає дані taffix. Всі зміни в розподіленому середовищі. Давайте подивимось його подробиці.

Це розподілена база даних або реєстр статусів, що зберігає дані про активи та їх переміщення / транзакції через мережу P2P [51. С. 1]. Кожна транзакція захищена паролем, а вся історія транзакцій групується та зберігається в блоці даних. Усі блоки закодовані та захищені і не можуть бути змінені. Кожна дія спричиняє створення незмінного запису транзакцій, що відбуваються в мережі.

Крім того, блок даних копіюється на кожен комп'ютер у мережі, тому кожен може отримати до нього доступ. Найбільша перевага блокчейну полягає в тому, що він може зберігати будь-який тип активів, а також всю детальну інформацію про ці активи, історію та розташування цих активів у мережі. Будь то цифрова валюта біткойн або будь-який інший цифровий актив, такий як сертифікати, особиста інформація, контракти, право власності на ІС чи навіть реальні об'єкти [51. С. 3].

Потужна функція блокчейну полягає в тому, що кожен користувач має певну мережеву потужність, оскільки кожен має власну копію мережі. Іншими словами, вузли, що беруть участь у мережі, не повинні знати або довіряти одне одному, оскільки кожен

може контролювати та перевіряти ланцюг сам. Як не дивно, взаємна недовіра інших користувачів - це те, що забезпечує безпеку блокчейну.

2.8.2. Валідація блоку

Як зазначалося вище, активи та їх транзакції зберігаються в мережі блокчейнів як блоки посилань. До мережі додаються лише дійсні транзакції. Технічно кажучи, верифікація блокчейну - це лише процес пошуку хешу блоку. У блокчейні всі блоки додаються до мережі лише після перевірки. Кожен раз, коли транзакція здійснюється в блокчейні, вона буде додана до блоку; іноді це одна транзакція в блоці, іноді це кілька транзакцій. Це залежить від розміру блоку та характеру мережі. Додаючи транзакцію до блоку, її потрібно спочатку перевірити та обробити, а потім додати до мережі блокчейнів як перевірений блок [51. С. 4-6]. Деякі алгоритми (наприклад, sha 256) можна використовувати для обчислення хеш-значення блоку.

Хеш-значення має певні властивості. Найголовніше, що хеш-значення не суперечить, тобто жоден два блоки не повинні мати однакового хеш-значення. Оскільки кожен блок представлений хеш-значенням, він повинен бути унікальним. Другий атрибут полягає в тому, що хеш-значення має бути незворотним. Це означає, що дані блоку неможливо отримати зі значення хешу.

2.8.3. Типи валідації блоку

Верифікатор блоку - це вузол, який бере участь у процесі верифікації блоку. Зусилля перевіряючого винагороджуються (насправді винагороджується їх обчислювальна потужність). Різні протоколи блокчейну використовують різні методи для вибору валідаторів з існуючого пулу вузлів. Розглянемо деякі з цих методів. метод роботи. Таким чином, проблема майнінгу відкрита для всіх. Усі майнери змагаються між собою, щоб додати наступний блок. Майнер, який знайшов рішення першим, отримає фіксовану винагороду. Насправді вузол з найбільшою обчислювальною потужністю зазвичай виграє гонку. Біткойн використовує алгоритм "робочого методу".

На цьому етапі досі невідомо, який метод є більш ефективним. Кожен спосіб має свої переваги та недоліки. Крім того, було введено багато інших методів для досягнення максимальної продуктивності мережі блокчейнів.

2.9. Проектування графічного інтерфейсу

Графічний інтерфейс системи повинен бути простим і зрозумілим, а всі кроки повинні бути інтуїтивно зрозумілими, щоб кожен, незалежно від вміння користуватися комп’ютером, міг легко освоїти систему. Графічний інтерфейс складатиметься з двох частин: перша система перегляду статистичних даних системи та результатів голосування, ця частина відкрита лише для системних адміністраторів; друга частина - усі користувачі можуть зареєструватися та голосувати в системі. На головному екрані

для перегляду статистики голосування повинні бути результати голосування, ідентифікатор голосування та дата. Принципова схема основного інтерфейсу наведена на рисунку 2.9..

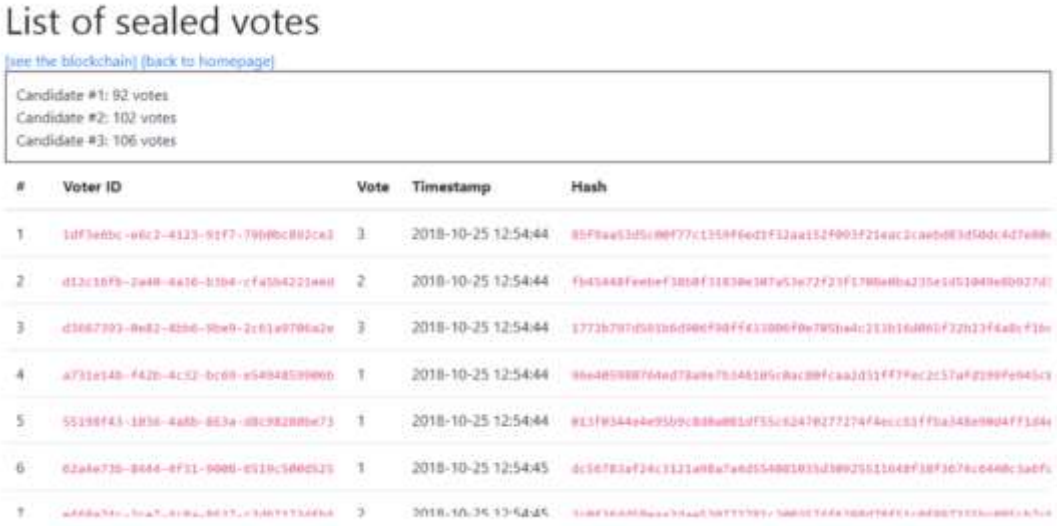

Рисунок 2.9 - Принципова схема інтерфейсу екрану результатів голосування

Крім того, ви можете відобразити сторінку, яка відображатиме інформацію про сам блокчейн і надаватиме можливість переглядати всі блоки в мережі. Інтерфейс вигляду системи показаний на малюнку. 2.10.

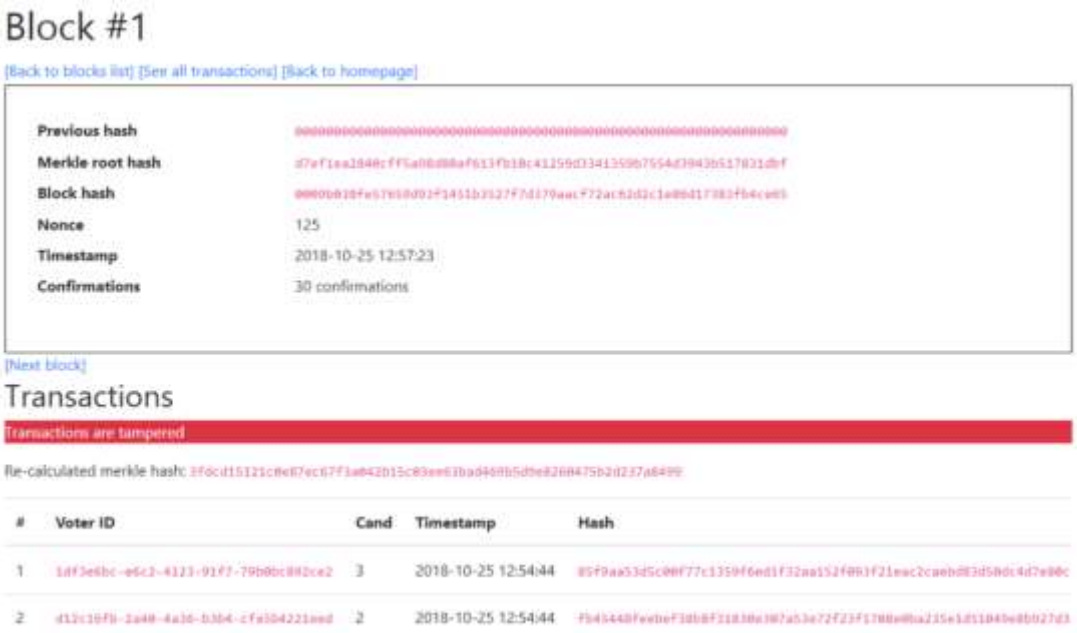

Рисунок 2.10 Схематичне зображення інтерфейсу екрану перегляду блоку з мережі блокчейн

Усі можливі кандидати повинні бути чітко відображені на головному екрані для голосування: вгорі буде форма для введення особистого ідентифікатора користувача, а

внизу - список кандидатів та форма для відбору кандидатів. Другий екран буде екраном для перегляду даних про кандидатів. Коли ви натискаєте ім’я кандидата на головному екрані, екран повинен відкритися. На рисунку 2.11 показано схематичну структуру інтерфейсу екрану для голосування.

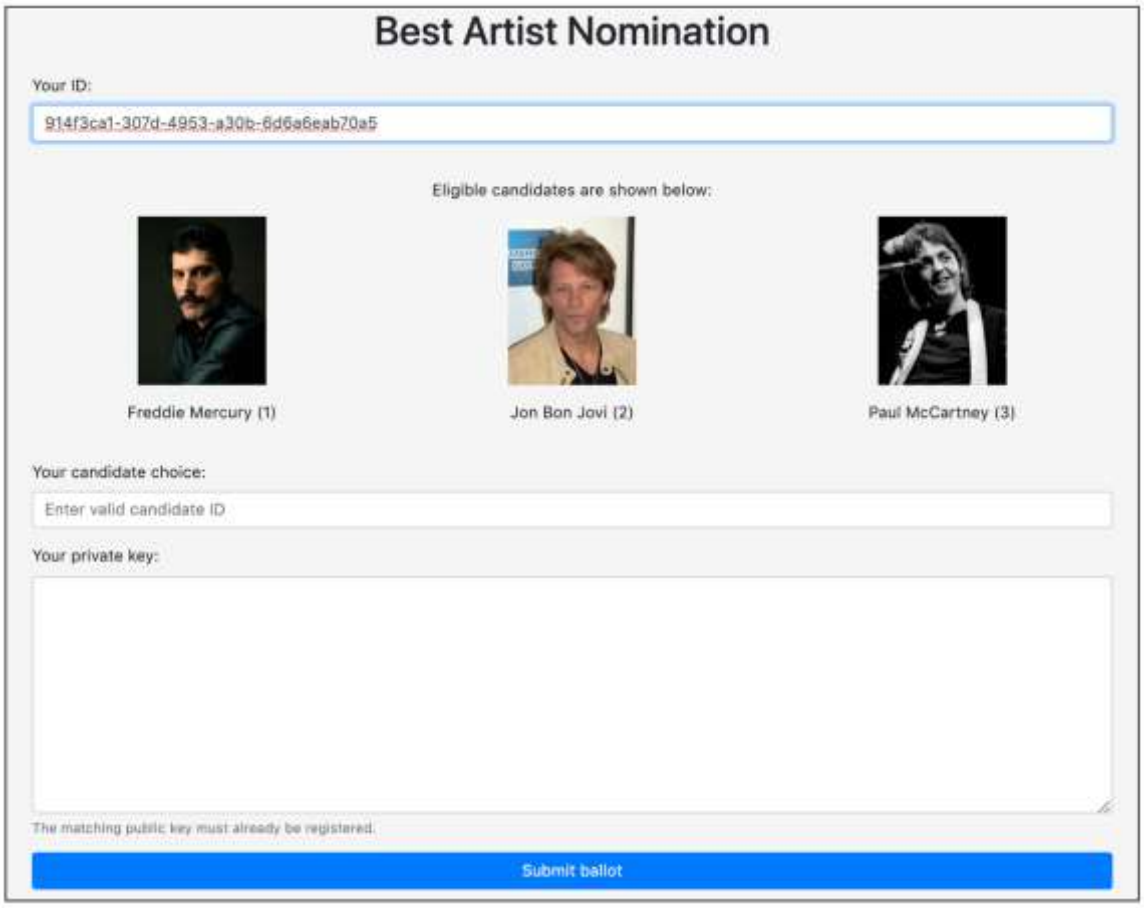

Рисунок 2.11 - Схематичне зображення інтерфейсу екрану голосування

ВИСНОВКИ ДО РОЗДІЛУ 2

У цьому розділі дається загальний опис предметних областей основних системних вимог та проблем формування функцій. Порівняно різні алгоритми досягнення консенсусу, розглянуто різні варіанти реалізації мережі блокчейнів та обрано найбільш підходящу реалізацію для завдання. Розглянемо різні способи використання системи. Для користувачів та адміністраторів заплановано дві частини системи. В результаті дослідження було побудовано графік альтернатив основних прецедентів. Перший приклад:

1) Реєстрація в системі;
2) Віддання голосу;
3) Отримання сповіщення.

Оскільки вони можуть мати різні варіанти розробки, залежно від попередніх операцій та поточного вибору користувача, вони були детально обговорені. Врахування цих особливостей на етапі проектування спростить процес розробки та уникне можливих помилок.

Також розроблений графічний інтерфейс користувача та графічний інтерфейс системного адміністратора.

РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ

3.1. Вибір технологій розробки систем з графічним інтерфейсом

3.1.1 Вибір мови програмування та інструменту для розробки графічного інтерфейсу користувача

Продуктивність програмування

Ефективність програмування визначає, як програміст з певним досвідом і знаннями може використовувати дану мову програмування для ефективного (тобто швидко та точно) вирішення поставлених перед ним завдань [51. С.111]. Порівняно з іншими загальними мовами програмування, функція Java та C # полягає у забезпеченні більш високої продуктивності програмування, ніж швидкість програми або ефективність пам'яті.

З цієї причини мови програмування Java та C # мають деякі додаткові функції. Наприклад, на відміну від C ++ (або C), програмісти не повинні явно «випускати» (виділену) виділену операційну систему виділеної пам'яті. Середовище виконання Java або C # автоматично звільняє невикористану пам'ять ("збір сміття"), що знижує продуктивність та ефективність пам'яті. Це звільняє програмістів від нудної роботи: відстеження випуску пам'яті - основного джерела програмних помилок. Порівняно з C ++ (або C), ця особливість Java та C # повинна значно покращити продуктивність процесу програмування [52. С. 112].

Ефективність використання пам'яті

У заяві про вимоги, наданій як не сценаріям, так і програмістам сценарію, сказано, що правильність єнайважливіший аспект для їх завдання. Однак оголошення, розміщене для програмістів сценаріїв (хоча це не опис вимог), також зробило більш широке завдання, згадуючи зусилля програмування,тривалість програми, читабельність програми / модуляризація / ремонтопридатність, елегантність рішення, споживання пам'яті та витрата часу роботи як критерії, за якими можуть оцінюватися програми[52. С. 114].

Це порівняння надає багато достовірності порівнянню робочого часу, показаному вище. Час, повідомлений для програмування сценаріїв, мабуть, лише помірковано занадто оптимістичний, якщо такий є, так що перевага робочого часу для мов сценаріїв приблизно фактора два має місце. Малюнок 20 показує ті самі дані, що і двовимірний графік, включаючи лінію регресії, яка може бути використана для (логарифмічно) прогнозування часу роботи від очікуваного розміру. Більш висока продуктивність мов сценаріїв відображається як лінія тренду, що лежить нижче на графіку[52. С. 115]. Рядок C / C ++ крутіший за інші, що в цьому логарифмічному графіку демонструє нелінійне збільшення зусиль: програми, які вдвічі довші, вимагають більше, ніж удвічі більше робочого часу. Можливо, це пов’язано з тим, що найкращі програмісти на C / C ++ не тільки були більш продуктивними, але й писали більш компактний код.

Ця різниця у фокусі, можливо, спрямувала трохи більше енергії на створення ефективної програми в Росіїгрупа сценаріїв порівняно з групою, що не виконується. З іншого боку, дві речі змогли це пом'якшитирізниця. Спочатку учасникам групи сценаріїв було прямо сказано: “Будь ласка, не переоптимізуйте свою програму.Надайте своє перше розумне рішення ”. По-друге, у групі без сценаріїв фільтрувались дуже неефективні програмиі відправлений назад для оптимізації в приймальному тесті[52. С. 116], оскільки тест накладає і час, і пам'ятьlimit1 відсутній у процедурі подання групи сценаріїв.Існувала ще одна різниця щодо приймальних випробувань та процедур вимірювання надійності: І те, і іншегрупи отримали невеликий словник (test.w, 23 слова) та невеликий файл входів (test.t) та правильні результати (test.out) для розробки програми та початкового тестування, а також великий словник (woerter2, 73113 слова) .Потім перевірку прийнятності для групи, що не виконується за допомогою сценарію, проводили за допомогою випадково створеного вхідного файлу (інакше кожного разу) та середньо-великий словник із 20946 слів[52. С. 117]. Невдалий прийомний тест коштував відрахування 10 Німецьких марок із загальної компенсації, виплаченої за успішну участь в експерименті, яка становила 50 Німецьких марок

(близько 30 доларів США).

На відміну від цього, групі сценаріїв було надано як вхідний файл z1000.in, так і відповідні правильні результати z1000.out, які використовуються для вимірювання надійності у цьому звіті і можуть виконати стільки тестів на них дані, як їм заманеться.

Володіння цими даними є, безперечно, перевагою для групи сценаріїв щодо необхідного робочого часу. (Примітка що прийомний тест у групі, що не є сценарієм, автоматично позначав і повідомляв про будь-які помилки окремо група сценаріїв повинна була виконати порівняння правильного результату та фактичного виводу самостійно. Веб-сторінка зазначив, що утиліти Unix для сортування та розрізнення можуть бути використані для автоматизації цього порівняння.) Більш серйозною проблемою є, мабуть, інший режим роботи: як уже згадувалося вище, багато сценарію [52. С. 114].

Учасники групи кілька днів думали над рішенням, перш ніж фактично його виготовити, тоді як учасники, які не писали сценарії, почали працювати над рішенням одразу після прочитання вимог. Це, мабуть, перевага для групи сценаріїв. Однак для більш ніж двох третин некриптової групи одна або кілька довше траплялись і робочі перерви (на ніч або навіть на кілька днів) [52. С. 115].

Підводячи підсумок, можна сказати, що завдання обох груп досить подібні, але будь-яке конкретне порівняння потрібно чітко приймати з достатньою кількістю солі. Можливо, була певна перевага для групи сценаріїв з повагою до умов роботи: деякі з них використовували неміряний час на роздуми перед фактичною роботою впровадження. Отже, слід покладатися лише на суттєві відмінності результатів. Садку це може призвести до зайвого блокування ресурсів, в гіршому — до порушення цілісності відкритих ресурсів[52. С. 117].

Порівняння AWT/Swing і Qt

На щастя, є спосіб, як ми можемо перевірити одразу дві речі, а саме правильність робочого часу звітність та еквівалентність можливостей програміста в сценарії порівняно з групою, що не є сценарієм. Зауважте, що обидві ці можливі

проблеми, якщо вони є, мають тенденцію зміщувати час роботи групи сценаріїв донизу: ми б очікуємо, що шахраї підробляють свій час меншим, а не більшим, і ми сподіваємось побачити більш здібних програмістів (а не менш спроможні) у групі сценаріїв у порівнянні з групою, яка не виконує сценарії, якщо можливості програміста відрізняються в середньому [52. С. 119]. Ця перевірка спирається на старе правило, яке говорить, що продуктивність програміста вимірюється в рядках коду на годину (LOC / година) приблизно не залежить від мови програмування: за кількома крайніми винятками таких як APL або Assembler, час, необхідний для кодування та тестування програми, часто визначається кількість функціональних можливостей, яка може бути виражена для кожного рядка, але час, необхідний для кожного рядка, буде приблизно постійним [52. С. 120].

Це правило в основному емпіричне, але його можна пояснити когнітивною психологією: коли програміст є досить вільно володіючи мовою програмування, один рядок коду є найважливішою одиницею мислення (принаймні під час фаз кодування та налагодження). Однак, якщо це домінуюче, обмеження ємності короткочасної пам'яті (7 одиниць плюс-мінус два) свідчить про те, що зусилля, необхідні для побудови програми, яка набагато довша за 7 рядки можуть бути приблизно пропорційні його кількості рядків, оскільки час, необхідний для правильного створення та обробка однієї одиниці є постійною і не залежить від обсягу інформації, представленої одиницею [53. С. 125].

Хоча оцінка API певною мірою залежить від особистих уподобань програміста, можна визначити ці API в API, що зробить розроблений код простішим, коротшим та зрозумілішим за інші API [53. С. 126]. Нижче наведено два приклади коду: перший використовує Java / Swing (рис. 2.1.1), другий використовує C ++ / Qt (рис. 2.1.2), який реалізує вставку декількох елементів у дерево ієрархії .

```
  ...
DefaultMutableTreeNode root = new DefaultMutableTreeNode( "Root" );
DefaultMutableTreeNode child1 = new DefaultMutableTreeNode( "Child 1" );
DefaultMutableTreeNode child2 = new DefaultMutableTreeNode( "Child 2" );
DefaultTreeModel model = new DefaultTreeModel( root );
JTree tree = new JTree( model );
model.insertNodeInto( child1, root, 0 );
model.insertNodeInto( child2, root, 1 );
...
```

Рис. 3.1. Приклад коду з використанням Swing

```
  ...
QListView* tree = new QListView;
QListViewItem* root = new QListViewItem( tree, "Root" );
QListViewItem* child1 = new QListViewItem( root, "Child 1" );
QListViewItem* child2 = new QListViewItem( root, "Child 2" );
...
```

Рис. 3.2. Приклад коду з використанням Qt

Як видно з малюнка, бібліотека Swing використовує архітектуру Model-View-Controller (MVC), і бібліотека Qt її підтримує, але використовувати її не обов’язково. Тому код, що використовує Qt, є більш інтуїтивним. Другий цікавий момент - це те, як різні інструменти пов'язують взаємодію користувача з певною функцією (яка називається функцією або методом). Наприклад, виберіть елемент у дереві вище (рис. 2.1.3 та 2.1.4). У синтаксисі Swing та Qt це виглядає по-різному, але основний принцип загальний.

```
  ...
tree.addTreeSelectionListener( handler );
...
```

Рисунок 3.3 - Приклад коду вибору елементу в дереві з використанням Swing

```
   ...
connect( tree, SIGNAL( itemSelected( QListViewItem* ) ),
         handler, SLOT( handlerMethod( QListViewItem* ) ) );
...
```

Рисунок 3.4 - Приклад коду вибору елементу в дереві з використанням Qt

Якщо врахувати конструкції, обрані авторами програм різними мовами, є разюча різниця.

Більшість програмістів у групі сценаріїв використовували асоціативні масиви, надані їхньою мовою, і зберігали слова словника для отримання за допомогою їх кодування чисел. Алгоритм пошуку просто намагається отримати з цього масиву, використовуючи в якості ключа префікси збільшення довжини решти решти поточного номера телефону [53. С. 129]. Будь-який знайдений збіг призводить до нового часткового рішення, яке буде завершено пізніше.

На відміну від них, фактично всі програмісти, які не виконують сценарії, обрали одне з наведених нижче рішень. У найпростішому випадку вони просто зберігають цілий словник у масиві, як правило, як у оригінальній формі символу, так і у відповідному поданні телефонного номера. Потім вони вибирають і тестують десяту частину цілого словника для кожної цифри телефонного номера, що кодується, використовуючи лише першу цифру як ключ для обмеження простору пошуку. Це призводить до простого, але неефективного рішення[53. С. 133].

Менша довжина програми сценаріїв може бути пояснена тим фактом, що велика частина фактичного пошуку виконується просто алгоритмом хешування, використовуваним всередині асоціативних масивів. Навпаки, програми без сценаріїв з їх реалізаціями у вигляді масивів або дерев вимагають, щоб більшість цих звичайних елементарних кроків процесу пошуку були явно закодовані програмістом [53. С. 135]. Це також виражається в зусиллях (або їх відсутності) для оголошень структур даних.

Цікаво відзначити, що, незважаючи на існування реалізацій хеш-таблиць як в бібліотеках класів Java, так і в C ++, жоден з програмістів, які не використовують скрипти, не використав їх (а, скоріше, реалізував деревоподібна рішення вручну), тоді як майже всі програмісти скриптів хеш-таблиці, вбудовані в мову, були очевидні вибір.

3.1.2 Бібліотека Qt

Qt — крос- платформна бібліотека розробки програмного забезпечення мовою програмування C++ (і не тільки), яка широко використовується для створення графічних інтерфейсів.

Більш складний випадок використовує 10-арне дерево, в якому кожен вузол представляє певну цифру, вузли на висоті n представляють n-й символ слова. Слово зберігається у вузлі, якщо шлях від кореня до цього вузла представляє числове кодування слова. Це найефективніше рішення, але воно вимагає порівняно великої кількості тверджень для реалізації конструкції дерева. У Java велика кількість об'єктів також призводить до великого споживання пам'яті через серйозні накладні витрати на пам'ять для кожного об'єкта, спричинені поточними реалізаціями мови [54. С. 155].

Через велику кількість реалізацій та широкий спектр досліджуваних програмістів, ці результати, взяті із залишком, є, мабуть, надійними, незважаючи на загрози обгрунтованості, розглянуті в Розділі 3. Однак слід підкреслити, що результати є дійсними для лише проблема телефонного коду, узагальнення для різних доменів додатків було б випадковим [53. С. 158].

Як видно на рис. 2.2.1, програми мають вигляд в різних операційних системах.

Рисунок 3.5 - Одне вікно в різних операційних системах

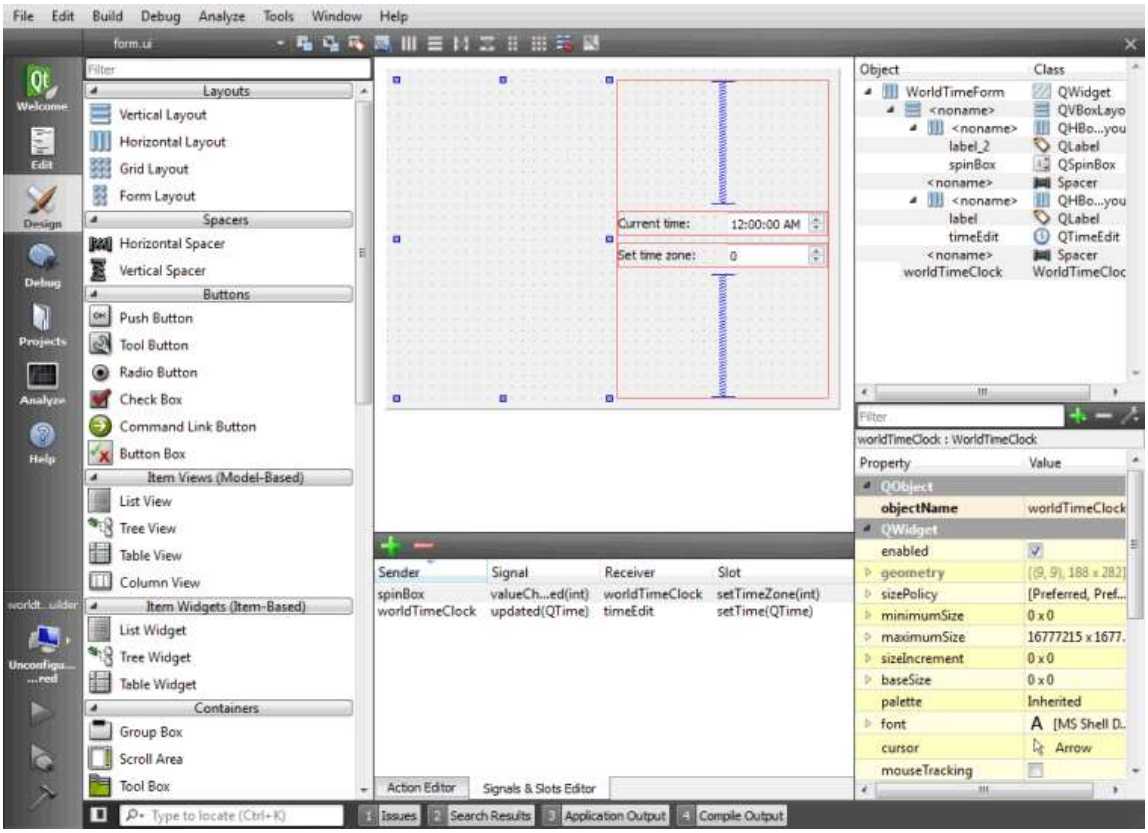

Рисунок 3.5 - Вікно редагування форми в Qt Designer Склад бібліотеки Qt

3.1.3. Вибір реалізації алгоритму консенсусу

Для реалізації блокчейн використаємо класичну модель яка використовується у криптовалюті Bitcoin. Для вирішення задачі досягнення консенсусу підходить алгоритм - Proof of work.

З моменту свого створення в 2009 році блокчейн біткойнов підживлює інновації, і ряд нових додатків, таких як смарт-контракти, були розроблені для використання переваг блокчейна. Біткойн був розгалужений кілька разів для точної настройки консенсусу (т. Е Часу генерації блоку і хеш-функції) і мережевих параметрів[50] (наприклад, розміру блоків і протоколу поширення інформації), а також для збільшення числа блоків. Ефективність. Наприклад, Litecoin і Dogecoin - найвідоміші Форк біткойнов - скорочують час генерації блоку з 10 до 2,5 і 1 хвилини [55. С. 135]. Паралельно з цими зусиллями з'явилися альтернативні децентралізовані мережі на основі блокчейнов (такі як Ethereum) з амбіціями щодо оптимізації консенсусу і мережі. параметри і спростити розгортання децентралізованих додатків поверх блокчейна.

На сьогодні існує багато реалізацій цього алгоритму, але так як програма буде використовувати мову програмування C++, то критерієм для вибору реалізації буде те, що бібліотека повинна мати підтримку цієї мови.

На блокчейни з підтримкою Proof of Work (PoW) в даний час припадає понад 90% загальної ринкової капіталізації існуючих цифрових криптовалюта. Хоча положення про безпеку біткойнов були ретельно проаналізовані, гарантії безпеки варіантних (розгалужених) ланцюжків блоків PoW (які були створені з різними параметрами) не отримали особливої уваги в літературі. Це відкриває питання, чи застосуємо існуючий аналіз безпеки PoW біткойнов до інших реалізацій, які були створені з іншими консенсусними і / або мережевими параметрами. У цій статті ми представляємо нову кількісну структуру для аналізу наслідків для безпеки і продуктивності різних узгоджених і мережевих параметрів ланцюжків блоків PoW. На основі нашої структури ми розробляємо оптимальні змагальні стратегії для подвійного витрачання та егоїстичного Майнінг, беручи до уваги обмеження реального світу, такі як поширення мережі, різні розміри блоків, інтервали генерації блоків, механізм поширення інформації та вплив атак затемнення [51. С.155]. Таким чином, наша структура дозволяє нам фіксувати існуючі розгортання на основі PoW, а також варіанти блокчейнов PoW, екземпляри яких створюються з різними параметрами, і об'єктивно порівнювати компроміси між їх продуктивністю і безпекою [51. С. 156].

Безпека PoW заснована на принципі, що жоден об'єкт не повинен збирати більше 50% обчислювальної потужності, тому що такий об'єкт може ефективно контролювати систему, підтримуючи найдовшу ланцюжок. Ми коротко опишемо відомі атаки на існуючі блокчейни на основі PoW. По-перше, зловмисник може спробувати двічі витратити, використовуючи одну і ту ж монету для проведення двох (або більше) транзакцій - таким чином, фактично витративши більше монет, ніж він має. Недавні

дослідження показали, що прийняття транзакцій без необхідності підтвердження блокчейна небезпечно [54. С. 156]. Чим більше підтверджень отримає транзакція, тим малоймовірно, що ця транзакція буде скасована в майбутньому. По-друге, Майнер можуть спробувати виконати атаки саморобного Майнінг [54. С. 256], щоб збільшити свою відносну частку Майнінг в блокчейне, вибірково утримуючи здобуті блоки. і тільки поступово їх публікуємо [54. С. 267]. Недавні дослідження показують, що в результаті цих атак самобутній майнер, спочатку оснащений 33% потужності Майнінг, може ефективно заробляти 50% потужності Майнінг. Атаки з подвійним витратою і саморобний Майнінг можуть бути зменшені, якщо всі вузли в системі блокчейн будуть тісно синхронізовані. Зверніть увагу, що крім затримки в мережі, затримки синхронізації можуть посилюватися через атак затемнення [54. С. 268], коли зловмисник створює логічний розділ в мережі, тобто надає таку, що суперечить інформацію про блоках і транзакціях різних вузлів мережі блокчейнов.

3.2. Основні рішення з реалізації системи та його компонентів

Для розробки системи в використовується архітектурний шаблон проектування модель-вигляд-контролер (Model-View-Controller, MVC).

В даний час, коли інформаційні технології є одним із найбільш швидко розвиваються напрямків бізнесу, багато компаній-розробники програмного забезпечення можуть з таким же якістю реагувати на вимоги клієнтів. Загалом, прикладне програмне забезпечення в основному складається з трьох великих модулів, таких як інтерфейс, бізнес-логіка і дані. У традиційних додатках між ними була дуже тісний зв'язок, тому на зорі розробки програмного забезпечення розробникам доводилося писати весь код для створення будь-якої програми. Тепер, з появою фреймворка MVC, вирішите зазначені вище проблеми. MVC (Модель-Представлення-Контролер) - це перша буква моделі, уявлення, контролера, розділяє вхідні дані програми, бізнес-логіку і вихідні дані відповідно до моделі, поданням і контролером [55. С.224]. MVC продемонстрував свої переваги для інтерактивних додатків, що дозволяють багаторазово представляти схожу інформацію, сприяючи повторному використанню коду і допомагаючи розробникам зосередитись на конкретній функції програми. Фреймворк MVC став стандартом в сучасній розробці програмного забезпечення. Відповідне повторне використання коду, що застосовується на рівні моделі, рівні уявлення і рівні управління, може не тільки привести до поділу базової бізнес-логіки, управління процесом і відображення, але також може значно поліпшити масштабованість програмного забезпечення і ремонтопридатність. Повторне використання коду - одна з найбільш поширених форм об'єктно-орієнтованого повторного використання. При розробці програмного модуля внутрішні модулі слід використовувати повторно в максимально можливій мірі. Повторне використання коду не тільки може значно прискорити швидкість розробки, скоротити інвестиції в розробку, але також може поліпшити якість системи, щоб полегшити концентрацію вирішення

проблем на етапі тестування. Переважна більшість розробників очікують повторного використання коду, що також є однією з цілей об'єктно-орієнтованого програмування. У цій статті описується швидка розробка простого генератора додатків для конкретного додатка [55. С.225].

Модель - уявлення - контролер (MVC) - це концепція архітектури програмного забезпечення, що розглядається як архітектурний патерн в розробці програмного забезпечення. Він складається з трьох компонентів, таких як модель, уявлення і контролер. На малюнку 1 показані відносини між компонентами архітектури MVC [55. С. 233]. У середовищі MVC уявлення і контролер належать призначеному для користувача інтерфейсу. Спочатку користувач відправляє запит контролеру через графічний інтерфейс користувача (GUI). Потім контролер доступу до моделі надає дані відповідно до запиту користувача. Після цього модель повертає дані контролеру, і контролер дані через зазначене подання. Ми також використовуємо базу даних для зберігання даних і надання вихідного джерела даних в нашій системі. Ми також резюмуємо наступні компоненти [55. С. 234]. На рис. 3.1.1 зображена діаграма взаємодії між трьома компонентами шаблону.

Рисунок. 3.6 - Діаграма взаємодії між компонентами шаблону MVC

Якщо об'єднати вид і контролер, то в результаті отримуємо архітектуру модель-вигляд (див. рисунок 3.7).

Рис. 3.7 - Архітектура модель-вигляд

Перший рівень складається тільки з користувачів. Користувач може відправити запит і отримати відповідь від компонентів «Перегляд» середнього рівня. По-друге, середній рівень складається з трьох компонентів з іменами Модель, Представлення і Контролер. Ці компоненти можуть обмінюватися даними і обробляти дані один з одним. Нарешті, третій рівень містить тільки базу даних, в якій дані зберігаються постійно. На малюнку 2 показано, як дані передаються по системі за допомогою MVC [55].

У цій параграфі ми представили java-фреймворк для швидкої розробки настільних додатків на основі MVC. Розробник програмного забезпечення може використовувати цю структуру для швидкого створення програмного забезпечення. Використовуючи цю структуру, не тільки досягається повне розділення уявлення, контролера і моделі режиму MVC, але також досягається поділ рівня бізнес-логіки й рівня уявлення. З нашого застосування для тестування ми вважаємо, що програмне забезпечення можна ефективно розробляти, правильно використовуючи цю структуру MVC, і ця структура MVC може бути одним з активних учасників спільноти розробників програмного забезпечення. Фактична експлуатація довела, що ця структура стабільна, ефективна і здатна розробляти високоякісні програми [56. С. 125-126].

3.2.1. Підключення усіх необхідних бібліотек

Так як більшість бібліотек являються вбудованими в мову програмування, окремо їх встановлювати непотрібно.

Також потрібно звернути увагу на ієрархію пакетів в бібліотеці.

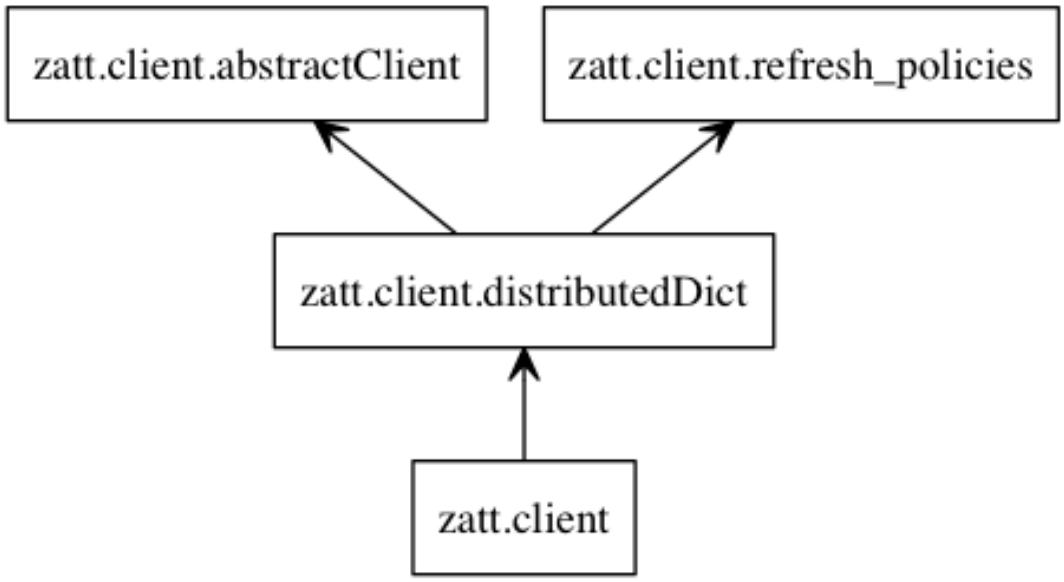

Рисунок 3.6 - Ієрархія модуля клієнта

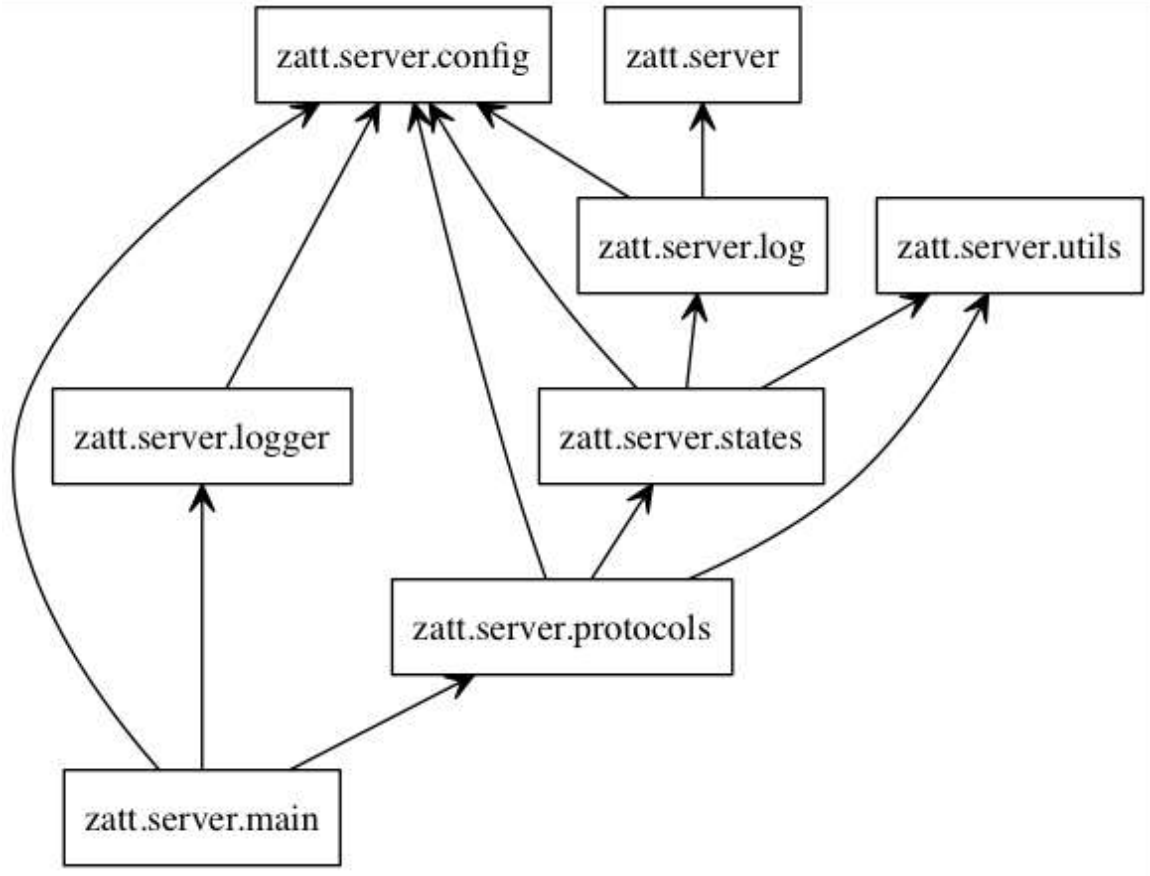

Рисунок 3.8 - Ієрархія модуля сервера

3.2.2. Реалізація структури блокчейн мережі

Для створення системи потрібно змоделювати основні компоненти, які використовуються у блокчейн. Такими компонентами є:

1. Блок;
2. Валідація блоку;
3. Хешування блоку.

Отже потрібно реалізувати прості моделі відповідних компонентів.
Представимо діаграми класів для цих компонентів системи на рисунках 3.5 і 3.6 .

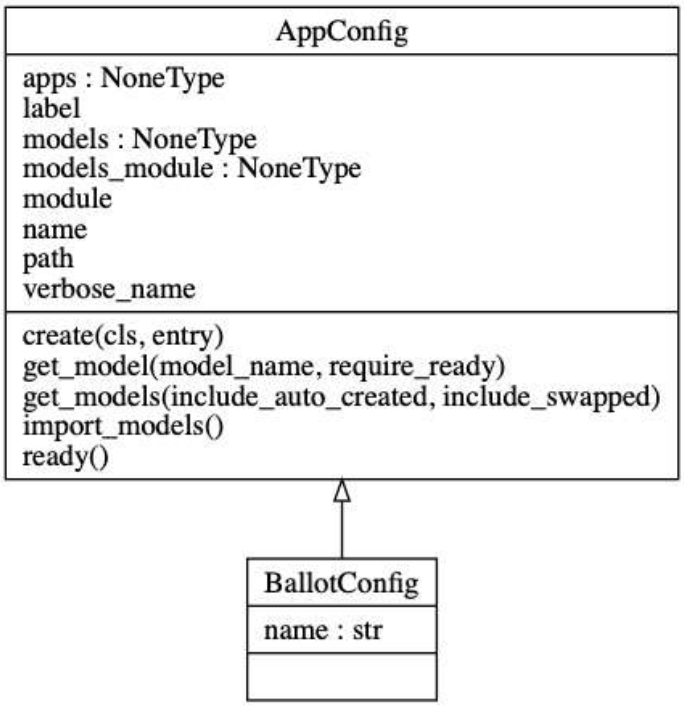

Рисунок 3.9 - Діаграма класів блоку

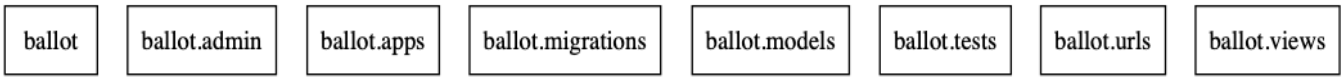

Рисунок 3.10 - Модулі пакету блоку

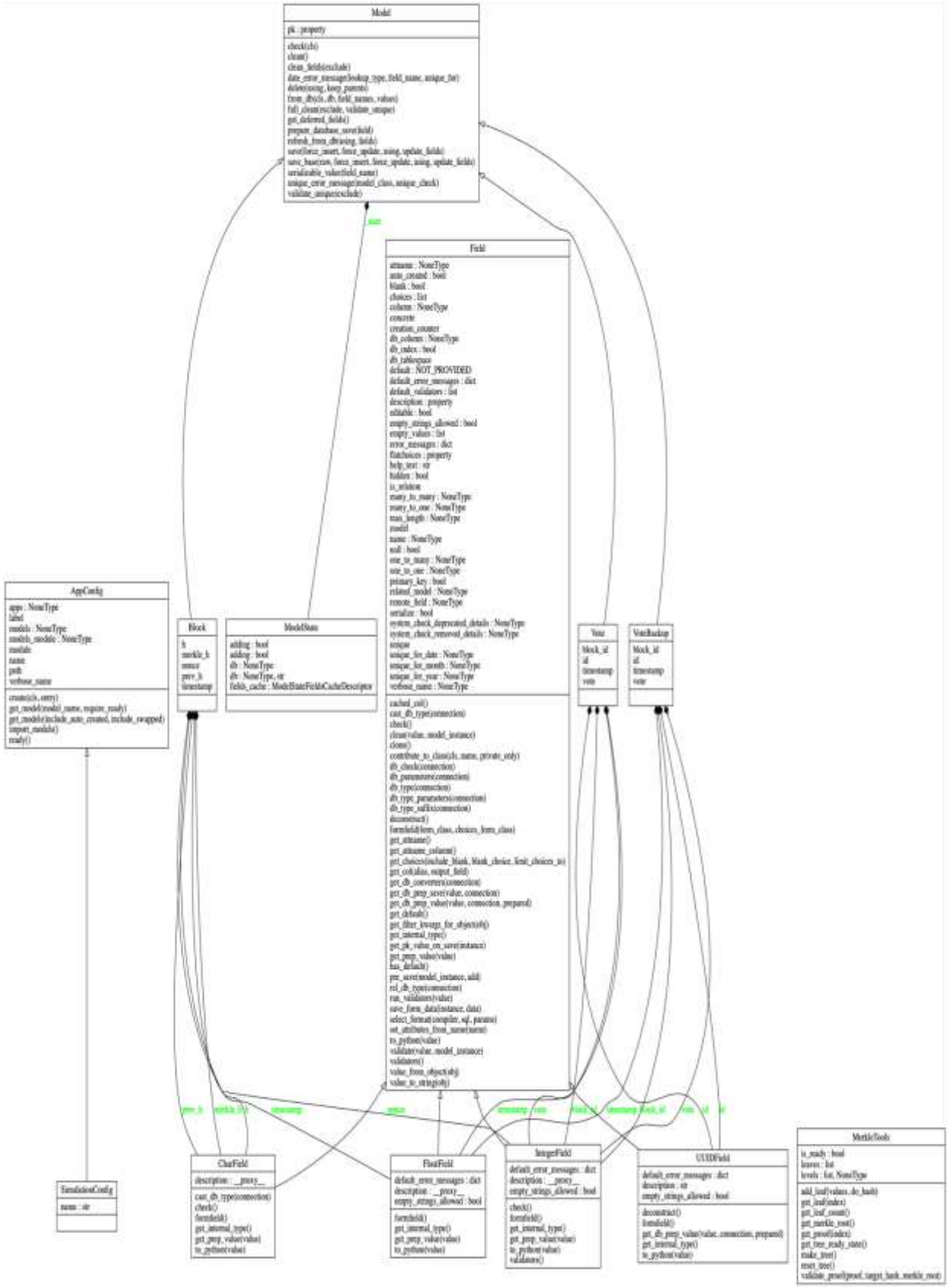

Рисунок 3.11 - Загальна діаграма класів модуляSimulation

3.2.3. Реалізація графічного інтерфейсу

Відповідно до спроектованого дизайну у додатку реалізовано два основних інферфейси, для користувача та для адміністратора: головний екран голосування та екран перегляду результатів голосування.

Головний екран голосування складається з декількох компонентів:

1) BallotForm – це компонент який містить форму, яка приймає голос користувача;
2) Seal – це компонент, що збирає дані про блок та валідує його печаттю;

Екран перегляду результатів складається з трьох компонентів:

3) NumOfVotes – це компонент для відображення кількості відданих голосів;
4) Block – це компонент, що показує дані по вибраному блоку;
5) BlockchainStat – це компонент для відображення всього блокчейну.

Для реалізації графічних компонентів скористаємося фреймворком Qt та мовою програмування QML. Даний фреймворк дозволяє створювати веб-застосунки дуже швидко, так як містить в собі багато реалізованих парадигм та патернів які використовуються у інтернеті, що значно пришвидшує та спрощує написання програми [].Проектування розподілених обчислювальних систем - це складний процес, що вимагає глибокого розуміння проблем проектування та теоретичних та практичних аспектів їх вирішення. Цей всебічний підручник висвітлює основні принципи та моделі, що лежать в основі теорії, алгоритмів та системних аспектів розподілених обчислень. Широке та детальне висвітлення теорії збалансовано практичними проблемами, пов'язаними із системами, такими як взаємне виключення, виявлення тупикових ситуацій, автентифікація та відновлення несправностей. Алгоритми ретельно відбираються, чітко подаються та описуються без складних доказів. Для з’ясування алгоритмів використовуються прості пояснення та ілюстрації.

ВИСНОВКИ ДО РОЗДІЛУ 3

У цьому розділі було розглянуто аспекти вибору стеку технологій та бібліотек для реалізації додатку, наведено обґрунтування вибору платформи для розробки системи, мови написання програми, показано рейтинг популярності операційних системи та мов програмування для швидкої розробки. Враховуючи вимоги до програмного продукту вибрано мову програмування та платформи, обрано допоміжні бібліотеки, які спростять процес розробки та дозволять побудувати гнучку архітектуру. Також розглянуті, можливі, варіанти реалізації мережі блокчейн, алгоритму консенсусу та кореня Меркла,

та було обрано бібліотеку std::hash з реалізацією криптографічних алгоритмів як найоптимальніший варіант.

Проведений опис основних рішень та підходів щодо реалізації проекту. Розроблено гнучку структуру додатку, яка проектувалась із розрахунком на можливе розширення, підхід для моделювання основних компонентів при налаштуванні з’єднання з базою даних та графічний інтерфейс для зручного використання системи.

РОЗДІЛ 4. РОЗРОБКА СТАРТАП-ПРОЕКТУ

a. Маркетинговий аналіз

Для того щоб зрозуміти потенційні можливості використання розробленого проекту потрібно провести маркетинговий аналіз. У наступних підпунктах буде наведено інформацію в таблицях по основним критеріям оцінки.

Таблиця 4.1 Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Голосування з використанням технології блокчейн	1. Державні вибори	Можливість швидко та безпечно проголосувати з комп'ютера або телефону
	2. Голосування за громадський бюджет	Можна проголосувати з будь-якого місця розташування
	3. Референдуми	

Аналогів та замінників на даний момент не існує, тому пункт про огляд та порівняння з конкурентами неможливий.

b. Технологічний аудит ідеї проекту

Таблиця 4.2 Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Голосування з	Блокчейн	Технологія	Доступна

	використанням технології блокчейн		наявна, потрібна реалізація за допомогою мови програмування	
2		C++	Технологія наявна	Доступна
3		Qt	Технологія наявна	Доступна
Обрана технологія реалізації ідеї проекту: мережа блокчейн з використанням мови програмування C++ та фреймворку Qt.				

З технічної точки зору реалізація проекту можлива, усі технології доступні та вже розроблені.

с. Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Таблиця 1.3 Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	1
2	Загальний обсяг продаж, грн/ум.од	невідомо
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Законодавчі обмеження

5	Специфічні вимоги до стандартизації та сертифікації	Наявні, система має бути сертифікованою
6	Середня норма рентабельності в галузі (або по ринку), %	15

Таблиця 4.4 Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Можливість волевиявлення-голосування	Усі групи населення	Не має значення	Простота використання та захищеність результатів

Цифрові стартапи, які запускають оригінальні ціннісні пропозиції, можуть протестувати і підтвердити свою бізнес-модель з використанням недавно з'явився набору практик, відомих як підходи до економічного запуску (LSA), які складаються з розвитку клієнтів і бережливого стартапу. Хоча LSA набирають обертів в екосистемі цифрових стартапів, вони як і раніше страждають від слабкої теоретичної бази і операційних проблем, які перешкоджають їх прийняття та впровадження. Мета цього дослідження - вийти за рамки згадування анекдотичних та одиничних прикладів і, за допомогою дослідження змішаних методів за участю 227 цифрових стартапів, надати перший великомасштабний аналіз: (i) чи застосовуються і як цифрові стартапи підходи до бережливим стартапам; (Ii) наступні результати; (Iii) основні переваги та недоліки, пов'язані з прийняттям і впровадженням LSA; і (iv) як цифрові стартапи з'єднують і комбінують LSA з іншими підприємницькими підходами та інструментами для запуску стартапів. Результати показують, що більша частина вибірки прийняла LSA і отримала ряд переваг від їх використання. Тому пропонується список практичних рекомендацій щодо усунення існуючих недоліків та підвищення ефективності прийняття і реалізації LSA. На закінчення пропонується основа для організації емпіричних висновків, в рамках яких АЛП вставляються в дебати теорії підприємництва по реалізації, підприємницькому бріколаж і створення можливостей. Потім пропонуються пропозиції

про те, як упорядкувати і зв'язати логіку реалізації та причинно-наслідкового зв'язку, а також інструменти прийняття рішень в «просторі підприємницьких можливостей».

Таблиця 4.5 Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Консервативне суспільство	Люди з консервативними настроями можуть бути проти електронного голосування	Інформаційна кампанія на телебаченні та радіо які показує усі переваги такого способу голосування
2	Застаріла законодавча база	Нинішні закони не дозволяють проводити електронне голосування	Зв’язатися з депутатами та запропонувати нові законопроекти

Таблиця 4.6 Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
11	Прогресивне суспільство	Люди з прогресивними настроями можуть допомогти у створенні системи та її тестуванні	Запуск тестових версій системи
22	Розвинена IT-спільнота	Розробники можуть взяти участь у	Відкриття нових вакансій

		створенні системи	

Таблиця 4.7 Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - чиста	Конкуренти відсутні	Розробити систему першими
2. За рівнем конкурентної боротьби - національний	Розробка актуальна лише для громадян України	Вийти на всесвітній ринок розробивши універсальну систему
3. За галузевою ознакою - міжгалузева	Голосування як процес, може відбуватися будь-де	Надати можливість створювати голосування людям самостійно
4. Конкуренція за видами товарів: - між бажаннями	Хто бажає може голосувати електронно	Популяризація електронного голосування через ЗМІ
5. За характером конкурентних переваг - нецінова	Користування системою для державних голосувань безкоштовне	
6. За інтенсивністю - не марочна	Конкуренти відсутні	

Можливість виходу на ринок досить висока, так як конкуренція на даний момент відсутня. Єдиним конкурентом можна вважати традиційне голосування. Отже фактори конкурентоспроможності можна навести в порівнянні з паперовим голосуванням.

Таблиця 4.8 Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних

		проектів значущим)
11	Простота проведення виборів	Майже нульові затрати на вибори, не потрібно друкувати бюлетені, не потрібно наймати людей для підрахунку голосів та спостерігачів
22	Швидкість	Вибори можна провести за кілька днів. Результати виборів будуть доступні відразу після завершення.
33	Залученість у процес	Проголосувати можна будь-де, маючи комп'ютер або телефон з підключенням до Інтернету

Таблиця 4.9 Порівняльний аналіз сильних та слабких сторін «Системи електронного голосування з використанням технології блокчейн»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з «Онлайн-голосування з блокчейн»						
			-3	-2	-1	0	1	2	3
1	Простота проведення виборів	3							+
2	Швидкість	6							+
3	Залученість у процес	9							+
4	Готовність суспільства	10					+		

4.4. SWOT-аналіз

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (табл. 4.10) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (табл. 4.9) [59. С. 127].

Таблиця 4.10 SWOT- аналіз стартап-проекту

Сильні сторони:простота виборів, захищеність результатів, швидкість виборів, залученість у процес, доступність	Слабкі сторони: певна технічна складність системи, потребує доступ до Інтернету на відміну від традиційного голосування
Можливості: економія коштів які витрачаються на організацію будь-яких виборів, більше людей зможе проголосувати, зменшення бюрократії	Загрози: застаріла законодавча база, неготовність та недовіра певних груп суспільства

Таблиця 4.11 (А) Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
11	Розробити систему та дозволити її використання для усіх бажаючих. Створення голосувань самостійно.	Висока	6 місяців

Таблиця 4.11 (Б) Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
22	Зробити систему закритою, з можливістю	Середня	1 рік

	створювати голосування лише державним органам.		

Так як система позиціонує себе як електронне голосування для усіх бажаючих провести таке голосування, і строк виконання такої системи менший то обираємо 1шу альтернативу.

Таблиця 2.12 Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
11	Усі	+	Високий	Конкуренція відсутня	Просто
Які цільові групи обрано: усі громадяни					

Таблиця 4.13 Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
11	Система голосування для усіх	Масовий маркетинг	Легкість та швидкість проведення чесних виборів	Стратегія Диференціації

Таблиця 4.14 Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
11	Так	Нових	Не буде	Стратегія Лідера

Таблиця 4.15 Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
11	Чесність та прозорість	Стратегія Диференціації	Легкість та простота проведення виборів.	Прозорі вибори Захищеність результату

	виборів. Захищеність результату. Легкість та доступність системи.			Швидкі вибори

4.5. Розроблення маркетингової програми стартап-проекту

Таблиця 4.16 (А) Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
11	Провести голосування	Швидкі та захищені вибори	Захищеність результату завдяки блокчейн технології

Таблиця 4.16 (Б) Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
22	Легко та просто проголосувати	Голосування зі смартфону або комп’ютера	Доступність системи онлайн
33	Голосувати з будь-якої точки	Голосування в мережі Інтернет	Можливість проголосувати з будь-якого місця

44	Заощадити кошти на проведенні голосування	Електронне голосування	Все електронно
55	Швидка реєстрація	Електронна реєстрація в системі голосування	Функція онлайн-реєстрації дозволяє голосувати, не вимагаючи посвідки на проживання тощо. Важливо мати лише документи про реєстрацію громадянина

Таблиця 4.17 Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Електронне голосування за допомогою технології блокчейн
II. Товар у реальному виконанні	Природа / Характеристика 1. Захистіть результат 2. Швидкість виборів 3. Ви можете голосувати в електронному режимі з будь-якого місця
	Якість: система розроблена з використанням передових технологій та сучасної мови програмування
	Пакування: цифровий продукт
	Марка: Система електронного голосування з використанням технології блокчейн

III. Товар із підкріпленням	До продажу: цифровий продукт
	Після продажу: цифровий продукт
Система реалізує заздалегідь визначені функції: можливе голосування, система контролюється адміністратором та можливість зберігати всі транзакції в блокчейні.	

Таблиця 4.18 Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
11	Невідомо	Невідомо	Безкоштовно для користувачів, система планує отримати державне фінансування	2 500 000 грн., 1 500 000 грн.

Таблиця 4.19 Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
11	Для потреб держави	Встановлення контакту, інформування, зберігання	Через Інтернет	Комбінована

Таблиця 4.20 Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
11	Економні споживачі	Усна, документна, електронна	Зробити вибори легким та швидким процесом.	Довести суспільству, що електронне голосування краще ніж традиційне	Офіційно-діловий стиль звернення

ВИСНОВКИ ДО РОЗДІЛУ 4

З результатів маркетингового аналізу можна зробити висновок, що проект є дуже перспективним, оскільки немає аналогів та конкурентів, і ця система має високий інтерес та попит на суспільство. Комерціалізація проекту стала можливою завдяки залученню державних та недержавних інвестицій.

Завдяки альтернативному розвитку, системі референдуму, проект може завоювати значну популярність. У повсякденному житті нам часто потрібно вибирати щось або когось, завдяки цьому проекту ми можемо використовувати мобільні телефони або комп’ютери, щоб безпечно виконати завдання. Ніхто не підозрює, що голосування "маніпулюють", оскільки кожен виборець використовує свій унікальний ідентифікатор та персональні дані, щоб ввести свої особисті дані, а потім проголосувати. Оскільки система використовує технологію блокчейну, ніхто не може підробити результати голосування.

Подальший розвиток проекту дуже зручний. Для повноцінної реалізації проекту потрібно домовитись з юристом щодо певних деталей, поспілкуватися з потенційними інвесторами проекту та отримати їхні відгуки.

ВИСНОВКИ

Ця магістерська робота присвячена дослідженню технології блокчейну та її застосуванню в електронному голосуванні. В ході роботи було розглянуто актуальність концепції програмного забезпечення. Вивчаються готові реалізації подібних систем, визначаються їх основні переваги та недоліки, і на основі цих даних перелічуються основні вимоги до системи. Порівнюються поняття звичайного, традиційного голосування та електронного голосування. Запропоновано можливий метод досягнення узгодженості даних у розподіленій системі. Порівняйте алгоритми консенсусу.

Провів дослідження в предметній області, визначив вимоги та завдання, які повинна вирішувати система, та склав список прецедентів та варіантів використання програми. Основні прецеденти будуть детально обговорені у можливих альтернативах. Дані структуровані та відображаються у вигляді таблиці. Відповідно до зазначених вимог проаналізуйте можливість використання існуючих технологій та платформ для реалізації системи. Вибрано мову прикладного програмування. Він також аналізує допоміжну бібліотеку, яка використовується для програмування, визначає та підтверджує її зручність.

В рамках системи реалізовано, заздалегідь визначений, функціонал: можливість віддати свій голос, моніторинг системи адміністратором, збереження усіх транзакцій у блокчейн.

Система реалізує заздалегідь визначені функції: можливе голосування, система контролюється адміністратором та можливість зберігати всі транзакції в блокчейні.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1) Технология блокчейн - то, что движет финансовой революцией сегодня / Дон Тапскотт, Алекс Тапскотт, 2018. – 30 - 36.
2) Blockchain Technology Explained: The Ultimate Beginner’s Guide About Blockchain Wallet / Alan T. Norman , 2017. – C. 43 – 45.
3) Blockchain Basics: A Non-Technical Introduction in 25 Steps / Daniel Drescher, 2017. – C. 29– 30, 85– 87.
4) Cryptoassets: The Innovative Investor's Guide to Bitcoin and Beyond / Chris Burniske, Jack Tatar, 2017. – C. 43 – 48.
5) The Basics of Bitcoins and Blockchains: An Introduction / Antony Lewis, 2018. – C. 112 – 115.
6) Mastering Blockchain: Deeper insights into decentralization, cryptography, Bitcoin, and popular Blockchain frameworks / Imran Bashir, 2017 .– C. 52 – 60
7) Blockchain: The Next Everything / Stephen P. Williams, 2019. – C. 87-92.
8) Blockchain For Dummies (For Dummies (Computer/Tech)) / Tiana Laurence, 2017, – C. 95 – 99.
9) Blockchain By Example: A developer's guide to creating decentralized applications using Bitcoin, Ethereum, and Hyperledger / Bellaj Badr, Richard Horrocks, Xun (Brian) Wu, 2018, – C. 241 – 255.
10) Blockchain: The Insights You Need from Harvard Business Review / Don Tapscott, Marco Iansiti, Karim R. Lakhani, Catherine Tucker, 2019, – C. 122 – 123.
11) Decentralized Voting: A Self-tallying Voting System Using a Smart Contract on the Ethereum Blockchain // Yang X., Yi X., Nepal S., Han F., 2018, – C. 256 – 262.
12) A Novel Approach to Implement Decentralized Voting System Using Blockchain / Tito Nadar, Mehul Rawal, Jay Patel, Abhishek Shah, A. S. Revathi, 2020, – C. 311 – 321.

13) he Cambridge Handbook of Smart Contracts, Blockchain Technology and Digital Platforms / Larry A. DiMatteo, Michel Cannarsa, Cristina Poncibò, 2019. – C. 101 – 107, 123 – 127, 213 – 217.
14) Legal Tech, Smart Contracts and Blockchain (Perspectives in Law, Business and Innovation) / Marcelo Corrales , Mark Fenwick, Helena Haapio, 2019. – C. 3 – 4.
15) Decentralized Applications: Harnessing Bitcoin's Blockchain Technology / Siraj Raval, 2016 – C. 15 – 23.
16) Mastering Bitcoin: Programming the Open Blockchain / Andreas M. Antonopoulos, 2017. – C.334 – 340.
17) Perspectives on the CAP Theorem / Seth Gilbert, Nancy A. Lynch, 2012 – C. 121 – 122 .
18) Introduction to Distributed Systems / Dr. Paul Sivilotti, 2007, C. 234–240.
19) Distributed Systems / Maarten van Steen, Andrew S. Tanenbaum, 2018. – 433-442 C.
20) Distributed Systems: An Algorithmic Approach / Sukumar Ghosh, 2011 – C. 121 – 122.
21) Information Security Technology for Applications: Internet Voting in Estonia / Vinkel P., 2011, – C. 80 – 82.
22) E-voting in Estonia:Technological Diffusion and Other Developments Over Ten Years / Mihkel Solvak, Kristjan Vassil, 2016, – C. 90 – 95.
23) Базы данных: проектирование, реализация и сопровождение / Томас Коннолли, Каролин Бегг, Анна Страчан, 2018, , – C. 90 – 95.
24) On the Security and Performance of Proof of Work Blockchains / Arthur Gervais, Ghassan Karame, 2016. – C. 234 –235.
25) Proof of Work and Proof of Stake consensus protocols: a blockchain application for local complementary currencies / Sothearath SEANG, Dominique TORRE, 2018. – C. 111.
26) Consensus Algorithms Third Edition / Gerardus Blokdyk, 2018. – C. 111.

27) Short gui consensus protocol [Електронний ресурс] – Режим доступу до ресурсу: https://www.coindesk.com/short-guide-blockchain-consensus-protocols .

28) Distributed Consensus Protocols and Algorithms/ Sachin S. Shetty; Charles A. Kamhoua; Laurent L. Njilla, 2019. – С. 111.

29) Consensus Algorithms: A Matter of Complexity? / Renato P. Dos Santos – , 2020, С. 55

30) Nonlinear and Dynamic Average Consensus Algorithms / Raman Jafroudi, 2006. – С. 97.

31) Consensus Algorithms / R. Indrakumari, T. Poongodi, Kavita Saini, B. Balamurugan, 2015. – С. 2-3.

32) Distributed Ledger Technology: The Science of the Blockchain / Roger Wattenhofer, 2018, С. 490-496.

33) The Bitcoin Standard: The Decentralized Alternative to Central Banking / Saifedean Ammous , 2016, С. 770-778.

34) he Internet of Money Volume Two: A collection of talks by Andreas M. Antonopoulos, 2018, С. 141–154.

35) That Book on Blockchain: A One-Hour Intro / Jonathan B. Morley, 2018, С. 335–350.

36) The Age of Cryptocurrency: How Bitcoin and Digital Money Are Challenging / Michael J. Casey, 2015, С. 316–317.

37) Build Your Own Blockchain: A Practical Guide to Distributed Ledger Technology / Arnd Huchzermeier , 2018, pp. 398–407.

38) The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology /Lamport, L. Tech.Rep. 2005.

39) Essentials of Blockchain Technology /Hai Jiang . 19, 2(2005), С. 78–103.

40) Mastering Ethereum: Building Smart Contracts and DApps / Gavin Wood Ph. D. (Author), 2019, С. 18–25.

41) Bitcoin, Blockchain, and Cryptoassets: A Comprehensive Introduction / Aleksander Berentsen (Author) (July1990), С. 463 – 492.

42) The Blockchain and the New Architecture of Trust (Information Policy) / Kevin Werbac, 2010, C. 13 – 15.

43) Cryptocurrency Investing Bible: The Ultimate Guide About Blockchain, Mining, Trading, ICO, Ethereum Platform, Exchanges, Top Cryptocurrencies for Investing and Perfect Strategies to Make Money / Alan T. Norman, 2011, C. 111.

44) Digital Gold: Bitcoin and the Inside Story of the Misfits and Millionaires Trying to Reinvent Money / Nathaniel Popper, 1989, C. 201–202.

45) Blockchain for Everyone: How I Learned the Secrets of the New Millionaire Class (And You Can, Too) / Sir John Hargrave (2012), C. 158.

46) Blockchain: Transforming Your Business and Our World / Philippa Ryan., 2020, C. 17-18.

47) Blockchain: A Practical Guide to Developing Business, Law, and Technology Solutions / Rene Madsen, 1992, C. 26–52.

48) Advances in Cryptology - CRYPTO '89: Proceedings / Brassard, Gilles, 2017, - C. 122-125.

49) ELECTRONIC SIGNATURES AND IDENTITIES: Law and Regulation / Lorna Brazell, 2015, C. 370–371.

50) CElectronic Signatures for B2B Contracts: Evidence from Australia / Aashish Srivastava, 2014, – C. 256-257.

51) Electronic Signatures: Authentication Technology from a Legal Perspective / M.H.M. Schellekens, – C. 212-215.

52) Understanding digital signatures / Grant, 2015, – C. 113-115.

53) Handbook of Applied Cryptography / Menezes, 2011. C. 50–51.

54) Cryptography and Coding 10th IMA International Conference, Cirencester / Smart, Nigel (Ed.), 2008, C. 19-22.

55) The Book of Qt 4: The Art of Building Qt Applications / Daniel Molkentin.-– 2014. – C. 456.

56) Getting Started with Qt 5: Introduction to Programming Qt 5 for Cross-platform / Baka, Benjamin, 2009, – C. 706.

57) An Introduction to Design Patterns in C++ with Qt 4 / Alan Ezust, Paul Ezust, 2004, – C. 568.

58) Programming with Qt / Matthias Kalle Dalheimer, 2016. – C. 511.